AD-A279 067

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
December 1989

3. REPORT TYPE AND DATES COVERED
Technical Oct 1989 - Dec 1989

4. TITLE AND SUBTITLE
A Mesh Generator for Finite Element Structural Analysis in Two Dimensions

5. FUNDING NUMBERS
C

6. AUTHOR(S)
Steven M. Buc

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
System Planning Corporation
1500 Wilson Blvd
Arlington, Virginia 22209

8. PERFORMING ORGANIZATION REPORT NUMBER
SPC 89-1465-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
1400 Wilson Blvd
Arlington, Virginia 22209

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Contract No. F2960-86-C-0026

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Finite element structural analysis is gaining wide acceptance in the field of projectile design. In the past, computer memory and storage requirements have limited the application of finite element analysis to large main frame computers and workstations. Recently, great improvements have been made in the computing power of desk top personal computers. As a result, interest has developed in applying finite element software to the P.C. environment. The mathematics and computer algorithms for finite element analysis are readily obtained from engineering references. However, the most tedious and time consuming operation is the generation of the finite element model. A suitable mesh generator is, therefore, required to fully implement any finite element package on a personal computer. This report presents a straight-forward and easy to implement two-dimensional mesh generator for both plane strain and axisymmetric analysis. Computer code listing and examples are provided.

14. SUBJECT TERMS
mesh generator, finite element analysis, projectiles, personal computer, FORTRAN

15. NUMBER OF PAGES
39

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

SPC 89-1465-12

A MESH GENERATOR FOR FINITE ELEMENT STRUCTURAL ANALYSIS IN TWO DIMENSIONS

DECEMBER 1989

Accesion For	
NTIS CRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

STEVEN M. BUC

94-14255

Prepared for:

The Defense Advanced Research Projects Agency
1400 Wilson Blvd
Arlington, Virginia 22209

Prepared by:

System Planning Corporation
1500 Wilson Blvd
Arlington, Virginia 22209

94 5 11 021

1. BACKGROUND

Finite element structural analysis is gaining wide acceptance in the field of projectile design. In the past, computer memory and storage requirements have limited the application of finite element analysis to large main frame computers and workstations. Recently, great improvements have been made in the computing power of desk top personal computers. As a result, interest has developed in applying finite element software to the P.C environment. The mathematics and computer algorithms for finite element analysis, in particular the extensive matrix operations required to solve the stiffness and deformation of the structure, have existed for many years, and are readily obtained from engineering references. However, the most tedious and time consuming operation is the generation of the finite element model. A suitable mesh generator is, therefore, required to fully implement any finite element package on a personal computer. This report presents an approach to developing the two-dimension finite element grid. The computer code listing is provided in the appendix, along with an example mesh generation. The documentation is sufficiently detailed to permit the programmer to tailor the mesh generation software to suit particular finite element analysis requirements.

2. MESH GENERATOR REQUIREMENTS

Advanced projectile structural design typically involves the analysis of complicated projectile shapes, such as the sabot, shown in cross-section, in Figure 2.1 Figure 2.2 shows a generic long rod penetrator within this structural sabot. This type of projectile configuration is typically used in kinetic energy anti-tank projectiles. The cross-section drawing shown in Figure 2.2 is created using a generic solids modeling algorithm. Table 2.1 shows the structural dimensions defining this generic projectile.

Table 2.1
Projectile Geometric Inputs

```
*
SABOT
2.0
   8   4
1.0000
 ELEM    D1        D2        L         Rho    FORM    R         H         K
   1    1.3       2.0       2.25               3
   2    2.0       2.5       0.25               43    0.5       0.2057    1.4557
   3    3.0       3.0       1.0                2
   4    2.5       2.0       0.25               44    0.5       0.2057    1.4557
   5    2.0       1.4       2.25               3
   6    1.4       1.65      0.5                43    1.0       0.0157    1.70
   7    1.65      3.0       1.0                3
   8    3.0       3.0       0.5                2
   9    1.0       1.0       6.5                2
  10    1.0       1.5       0.25               42    .5        0.2057    -0.2943
  11    1.5       2.7       0.85               3
  12    2.7       2.7       0.40               2
*
PEN
0.0
   1   0
6.6400
 ELEM    D1        D2        L         Rho    FORM    R         H         K
   1    1.0       1.0       10.5               2
**
```

Figure 2.1
Cross-Section of a Generic Projectile Sabot

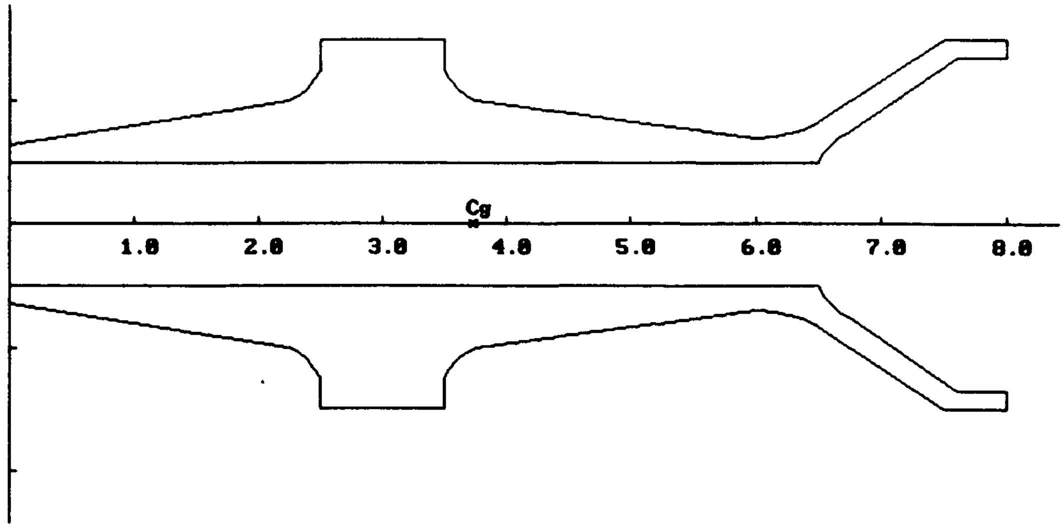
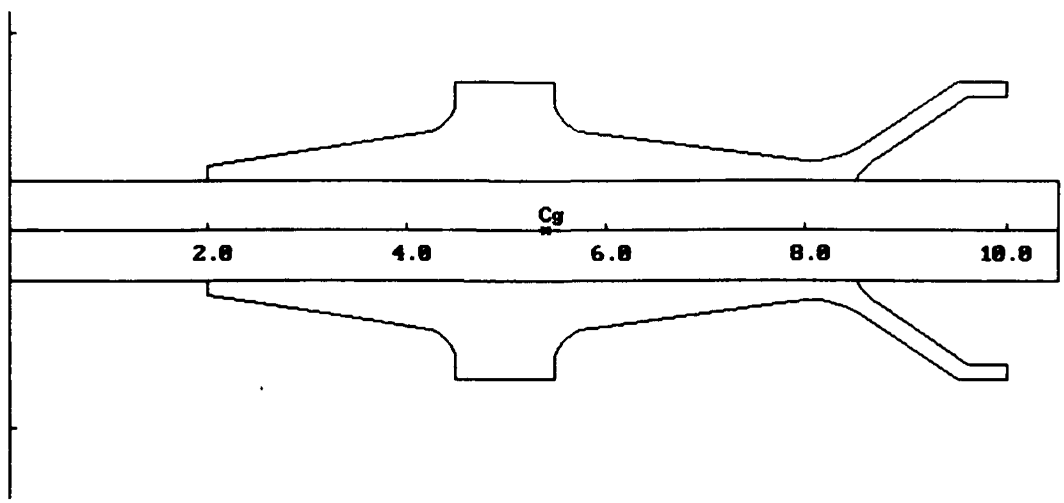


Figure 2.2
Cross-Section of Generic Kinetic Energy Projectile

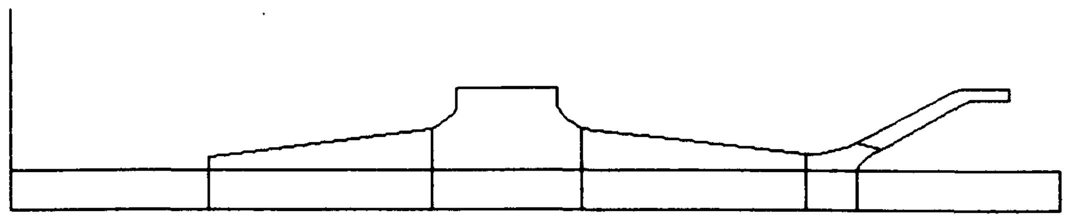


What is desired is to create a finite element grid within the boundaries of this projectile form. Finite element analysis typically requires the structure to be divided into individual quadrilateral elements. By convention, each element is defined by four corner nodes, in either a clock-wise or counter clock-wise direction, depending on the finite element algorithm. To ensure that all of the elements are structurally connected, adjacent elements are defined by common node numbers, depending on their location in the assembled grid. A file structure is then developed which allows the finite element analysis program to create the stiffness matrix, based on the element corner nodes and the x and y or z and r coordinates of each node.

3. CREATING THE FINITE ELEMENT GRID

The mesh generator algorithm presented in this report follows a systematic process to creating the finite element grid. First the projectile form is subdivided into homogenous areas, by defining area corner points and lines, based on the modeler's vision of how the grid is to be developed. Figure 3.0 shows an outline of the structure, when broken into sub-areas for grid generation. Figure 3.1 shows 32 discrete points defining the boundary of this structure. Table 3.1 shows the input format for these 32 points, as required by the software. Four additional points are entered in this example, which are not corner points, but are required for complete description of the geometry. The structure in Figure 2.2 includes four radii, which describe the curvature of the outline at the base of the central bulkhead and the root of the front scoop. For each radius in the structure, the center point of the curvature also needs to be defined. Therefore, points 12, 15, 22, and 26 are the centers for these curves. Each point input defines the point number, followed by its x and y position for a two-dimension plane strain model, or its z and r position, if the model is axisymmetric. For this example, the projectile is an axisymmetric structure, so the coordinates are in terms of the longitudinal position and the radial position of the point.

Figure 3.0
Geometry Sub-Area Definitions



Following the corner point definitions, the perimeter lines for each of the sub-areas are defined. Table 3.2 gives the line inputs for this example, and Figure 3.2 shows the line definitions graphically. Each line input defines the line number, followed by the two line end points, the number of finite divisions for this line, and the center point if the line is a radius. When defining the lines, as when defining the sub-areas and their corner points, the modeler needs to have a scheme in mind for how the mesh is to be generated. Line divisions should be consistent for opposite sides of each sub-area, so that a consistent quadrilateral grid will be generated. In addition, the line divisions for adjacent areas should be consistent so that the entire structure can be merged together.

Table 3.1
Sub-Area Corner Point Inputs

```
P,1,0,0.0
P,2,0,0.5
P,3,2.0,0.0
P,4,2.0,0.5
P,5,4.25,0.0
P,6,4.25,0.5
P,7,4.25,1.0
P,8,2.0,0.65
P,9,5.75,0.0
P,10,5.75,0.5
P,11,4.5,1.25
P,12,4.0443,1.4557
P,13,4.5,1.5
P,14,5.5,1.5
P,15,5.9557,1.4557
P,16,5.5,1.25
P,17,5.75,1.0
P,18,8.0,0.0
P,19,8.0,0.5
P,20,8.0,0.7
P,21,8.5,0.825
P,22,8.0000,1.7
P,23,8.5,0.0
P,24,8.5,0.5
P,25,8.75,0.75
P,26,8.9557,0.2943
P,27,9.6,1.35
P,28,9.5,1.5
P,29,10.0,1.35
P,30,10.0,1.5
P,31,10.5,0.0
P,32,10.5,0.5
```

Figure 3.1
Points Defining the Sub-Areas of the Structure

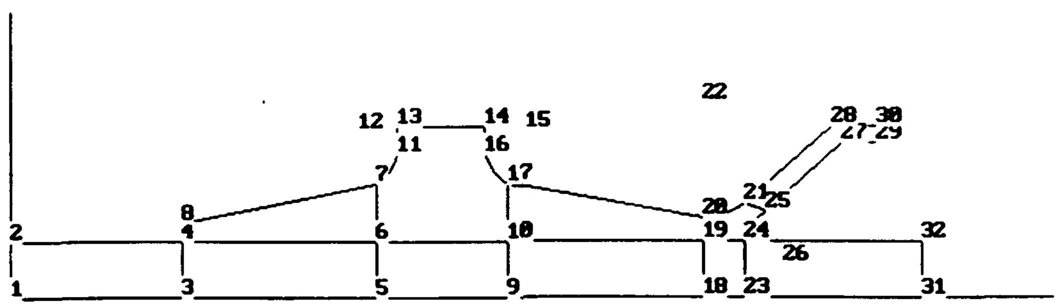


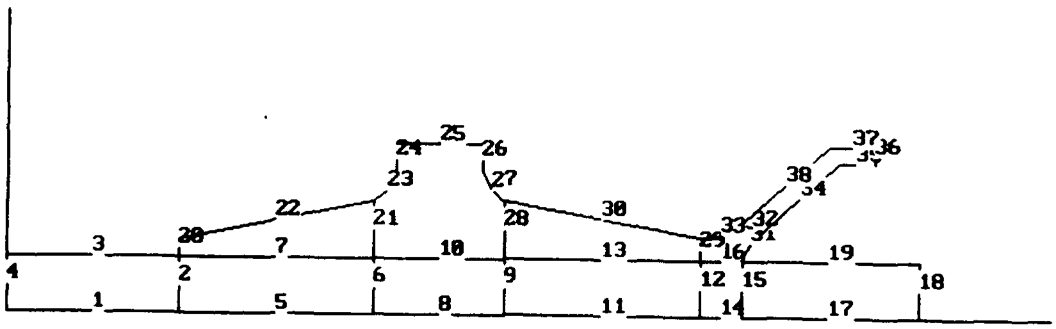
Table 3.2

Sub-Area Line Inputs

```
L,1,1,3,5,,
L,2,3,4,5,,
L,3,4,2,5,,
L,4,1,2,5,,
L,5,3,5,20,,
L,6,5,6,5,,
L,7,6,4,20,,
L,8,5,9,20,,
L,9,9,10,5,,
L,10,10,6,20,,
L,11,9,18,20,,
L,12,18,19,5,,
L,13,19,10,20,,
L,14,18,23,8,,
L,15,23,24,5,,
L,16,24,19,8,,
L,17,23,31,4,,
L,18,31,32,5,,
L,19,32,24,4,,
L,20,4,8,5,,
L,21,6,7,5,,
L,22,7,8,20,,
L,23,7,11,4,12
L,24,11,13,3,,
L,25,13,14,20,,
L,26,14,16,3,,
L,27,16,17,4,15
L,28,10,17,5,,
L,29,19,20,5,,
L,30,20,17,20,,
L,31,24,25,5,26
L,32,25,21,3,,
L,33,21,20,5,22
L,34,25,27,10,,
L,35,27,29,3,,
L,36,29,30,3,,
L,37,30,28,3,,
L,38,28,21,10,,
```

Figure 3.2

Lines Defining the Sub-Areas of the Structure



One may have noticed that three of the sub-areas are defined by more than four lines and more than four perimeter points. Since typical structures involve complicated shapes, which are not easily described by four nearly parallel sides, mesh generation requires the use of splines. Splines are groups of lines, which are to be treated as a single line in the mesh generation process. This approach makes mesh generation of complicated shapes easier to accomplish. Table 3.3 shows the five spline inputs for this example. When defining splines, at most five lines can be connected, and they should be listed in the order they appear in the geometry. As with the definition of individual lines, the total number of divisions for a spline should be consistent with the opposite side of the sub-area which it describes.

Table 3.3
Spline Inputs

```
S,1,21,23,24,,,
S,2,28,27,26,,,
S,3,32,33,,,,
S,4,34,35,,,,
S,5,37,38,,,,
```

Following the spline inputs, the model is ready to be meshed with a finite element grid. The mesh commands define each sub-area to be meshed in accordance with the points, lines, and splines described earlier. For each area, the four corner points are entered in either a clock-wise or counter clock-wise direction, depending on the convention used in the finite element software. Table 3.4 shows the area mesh commands used in this example. The command line consists of the area number, followed by the four corner points, followed by the mesh technique (1 or 2), and finally the material number, so that material properties can be discriminated later in the finite element analysis. Two mesh technique numbers were developed in the software, since any one technique is not always the best, depending on the shape of the sub-area. Technique 1 works well for areas which are more square or rectangular, as opposed to areas which include a curved perimeter. Some experimentation is required to develop the best looking grid for a particular sub-area. The algorithm is adequately general to permit the grid to take several forms. Results

will vary depending on the direction and the order used for the area corner points in the mesh command. The modeler needs to decide which results are best for the particular application. Figure 3.3 shows the resulting mesh using the commands in Table 3.4. Figure 3.4 shows a close-up of the mesh in the bulkhead region. Figure 3.5 shows a close-up of the root of the front scoop with a point and line plot. Figure 3.6 is the resulting mesh for the front scoop. The mesh generation for both of these areas included the use of splines.

Table 3.4
Area Mesh Command Inputs

```
A,1,3,4,2,1,1,1
A,2,3,4,6,5,1,1
A,3,5,9,10,6,1,1
A,4,9,18,19,10,1,1
A,5,18,23,24,19,1,1
A,6,23,31,32,24,1,1
A,7,4,6,7,8,1,2
A,8,6,10,14,13,1,2
A,9,10,19,20,17,1,2
A,10,20,25,24,19,2,2
A,11,29,30,21,25,1,2
*
```

Figure 3.3
The Complete Finite Element Grid

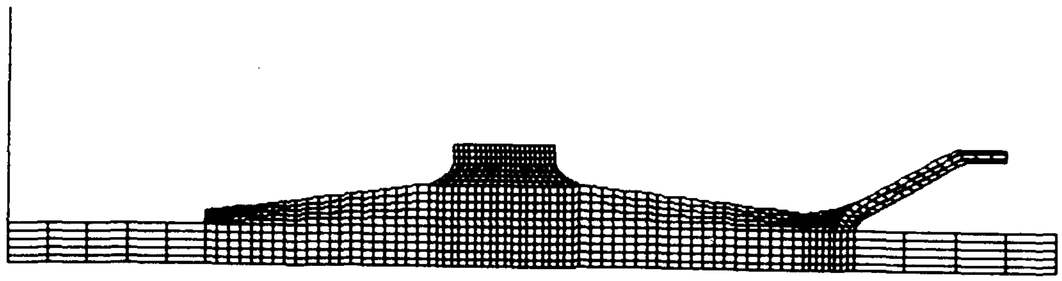


Figure 3.4
Close-Up of the Bulkhead Region Grid

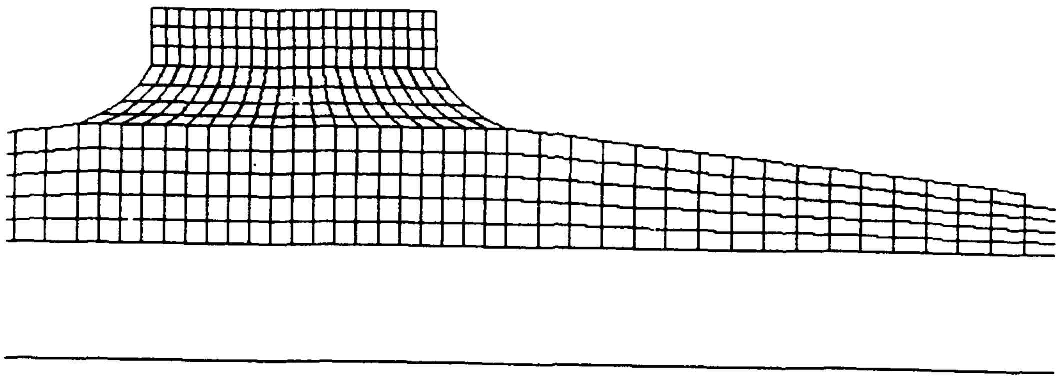


Figure 3.5
Point and Line Plot of the Front Scoop Region

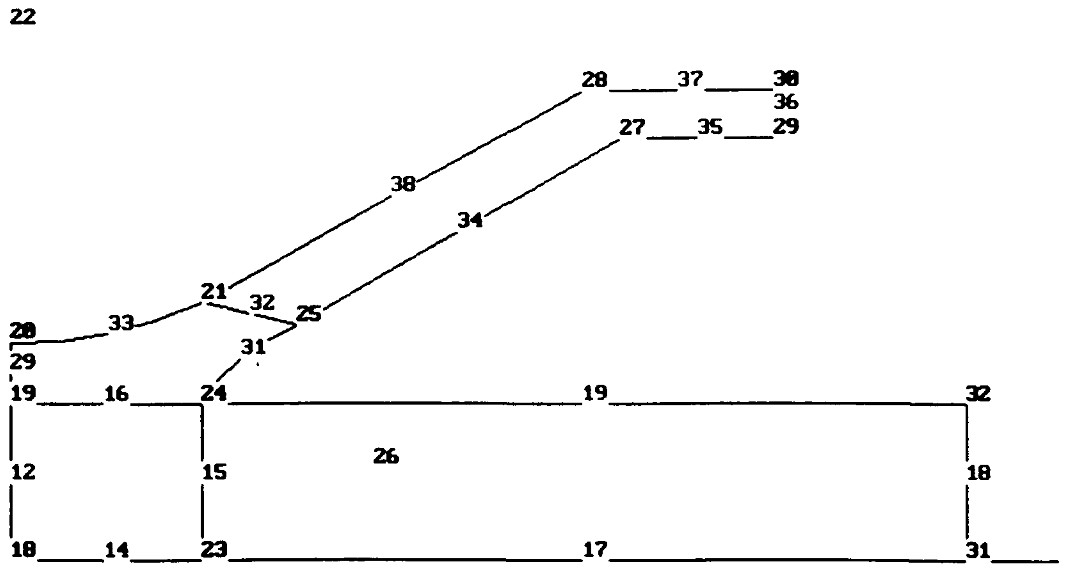
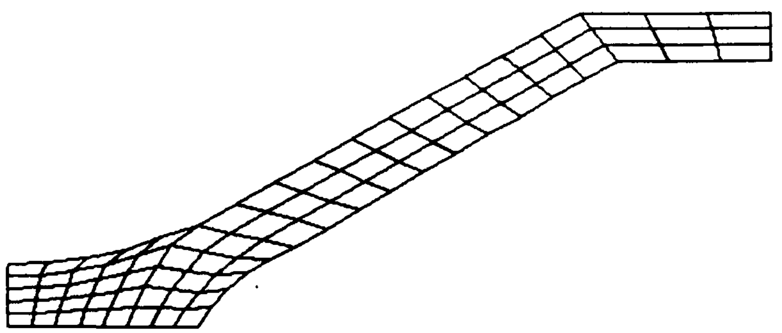


Figure 3.6
Front Scoop Region Grid



4. DETAILS OF THE MESH GENERATION PROCESS

For each sub-area mesh command, the mesh generation process begins by identifying the line or spline which connects the first two points in the command line. Since the line or spline could have been defined in the opposite direction, a test is performed on each line to see if it is reversed. If the two points define a spline, each line in the spline is identified as well. The meshing process begins by subdividing all the edge lines or splines according to the number of divisions in the line definitions. Several scratch files are also maintained to keep track of the line division node numbers for future reference, particularly when merging adjacent areas. The nodes defined on the first line of the area are then swept across the width of the area based on the division spacing of the line or spline between the second and third points. This where the algorithm uses either technique 1 or 2 when generating the grid. If technique 1 is used, the nodes sweep straight across to the opposite side. With technique 2, the curvature of the second side adjusts the node spacing as they sweep across. Within each sub-area all elements are connected by adjacent nodes according to the number of divisions on the perpendicular sides, but adjacent areas are not yet connected. Following completion of the meshing of all sub-areas, the operator can merge perimeter nodes, so that a solid structure is created. Merging is accomplished by cycling through all sub-areas to identify those which happened to have common boundary lines. Scratch files, created previously, contain the edge nodes for each line. The merge routine selects the lowest node number and makes it common to all elements connecting at that location. After merging, some node numbers will no longer be in use. The operator can then select the compress command, which will identify all un-used nodes in the node array. Un-used nodes are then re-assigned to elements having higher node definitions. This command is useful in ensuring that the size of the stiffness matrix is minimized in the finite element analysis. Figure 4.1 shows a close-up of the point and line plot of the left end of the example structure. Figure 4.2 shows the node and element definitions at the vertical sub-area boundary. At this boundary, the nodes have been merged so that the elements are structurally connected.

Figure 4.1
Close-Up of the Left End of the Example Structure

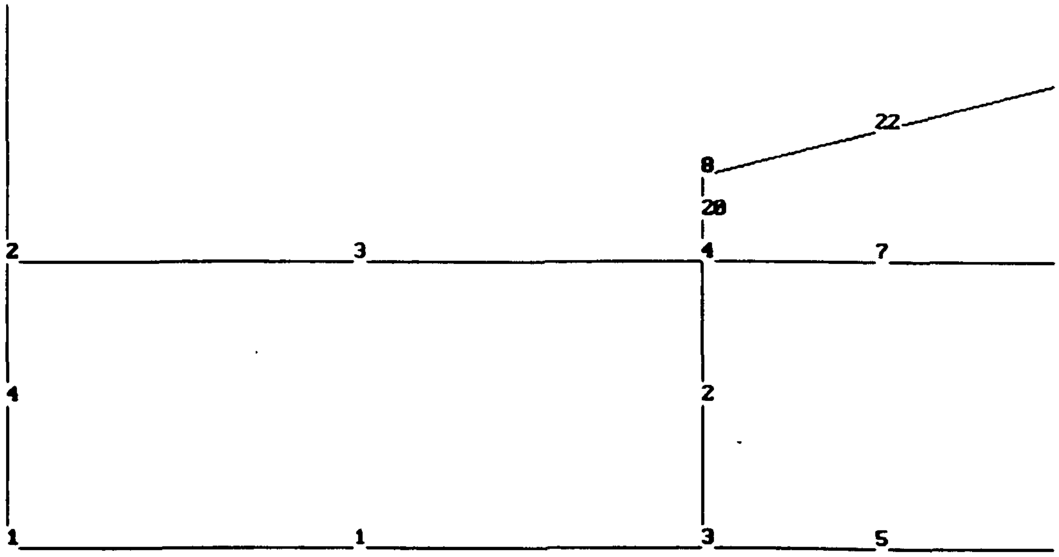
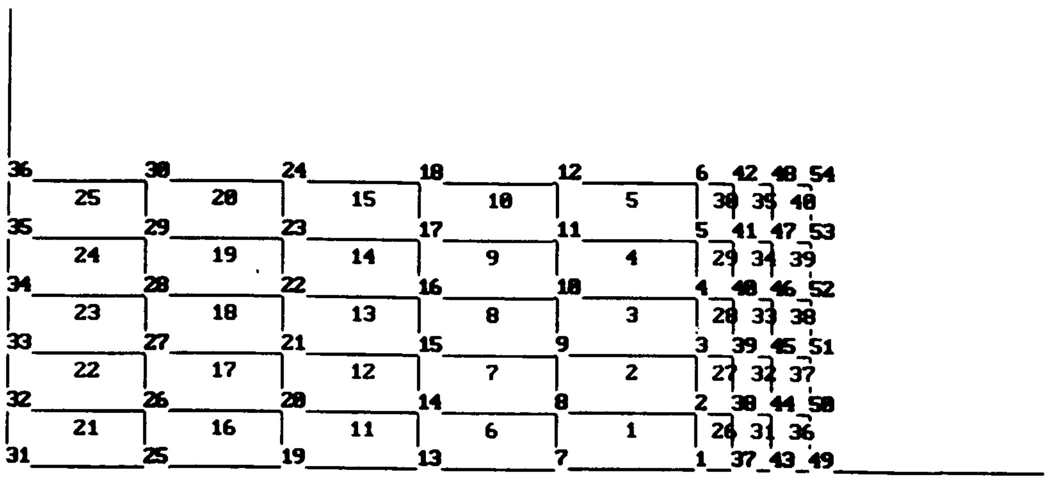


Figure 4.2
Merged Node and Element Plot at the Sub-Area Boundary



APPENDIX A
EXAMPLE INPUT AND OUTPUT

A.1 -- INPUT FILE
A.2 -- NODE FILE OUTPUT
A.3 -- ELEMENT FILE OUTPUT

A.1 -- INPUT FILE

```
P,1,0,0.0
P,2,0,0.5
P,3,2.0,0.0
P,4,2.0,0.5
P,5,4.25,0.0
P,6,4.25,0.5
P,7,4.25,1.0
P,8,2.0,0.65
P,9,5.75,0.0
P,10,5.75,0.5
P,11,4.5,1.25
P,12,4.0443,1.4557
P,13,4.5,1.5
P,14,5.5,1.5
P,15,5.9557,1.4557
P,16,5.5,1.25
P,17,5.75,1.0
P,18,8.0,0.0
P,19,8.0,0.5
P,20,8.0,0.7
P,21,8.5,0.825
P,22,8.0000,1.7
P,23,8.5,0.0
P,24,8.5,0.5
P,25,8.75,0.75
P,26,8.9557,0.2943
P,27,9.6,1.35
P,28,9.5,1.5
P,29,10.0,1.35
P,30,10.0,1.5
P,31,10.5,0.0
P,32,10.5,0.5
L,1,1,3,5,,
L,2,3,4,5,,
L,3,4,2,5,,
L,4,1,2,5,,
L,5,3,5,20,,
L,6,5,6,5,,
L,7,6,4,20,,
L,8,5,9,20,,
L,9,9,10,5,,
L,10,10,6,20,,
L,11,9,18,20,,
L,12,18,19,5,,
L,13,19,10,20,,
L,14,18,23,8,,
L,15,23,24,5,,
L,16,24,19,8,,
L,17,23,31,4,,
L,18,31,32,5,,
L,19,32,24,4,,
L,20,4,8,5,,
L,21,6,7,5,,
L,22,7,8,20,,
L,23,7,11,4,12
L,24,11,13,3,,
L,25,13,14,20,,
L,26,14,16,3,,
L,27,16,17,4,15
L,28,10,17,5,,
L,29,19,20,5,,
L,30,20,17,20,,
L,31,24,25,5,26
L,32,25,21,3,,
L,33,21,20,5,22
L,34,25,27,10,,
L,35,27,29,3,,
L,36,29,30,3,,
L,37,30,28,3,,
L,38,28,21,10,,
S,1,21,23,24,,,
S,2,28,27,26,,,
S,3,32,33,,,,
S,4,34,35,,,,
S,5,37,38,,,,
A,1,3,4,2,1,1,1
A,2,3,4,6,5,1,1
A,3,5,9,10,6,1,1
A,4,9,18,19,10,1,1
A,5,18,23,24,19,1,1
A,6,23,31,32,24,1,1
A,7,4,6,7,8,1,2
A,8,6,10,14,13,1,2
A,9,10,19,20,17,1,2
A,10,20,25,24,19,2,2
A,11,29,30,21,25,1,2
*
```

A.2 -- NODE FILE OUTPUT

1	2.00000	0.000000
2	2.00000	0.100000
3	2.00000	0.200000
4	2.00000	0.300000
5	2.00000	0.400000
6	2.00000	0.500000
7	1.60000	0.000000
8	1.60000	0.100000
9	1.60000	0.200000
10	1.60000	0.300000
11	1.60000	0.400000
12	1.60000	0.500000
13	1.20000	0.000000
14	1.20000	0.100000
15	1.20000	0.200000
16	1.20000	0.300000
17	1.20000	0.400000
18	1.20000	0.500000
19	0.800000	0.000000
20	0.800000	0.100000
21	0.800000	0.200000
22	0.800000	0.300000
23	0.800000	0.400000
24	0.800000	0.500000
25	0.400000	0.000000
26	0.400000	0.100000
27	0.400000	0.200000
28	0.400000	0.300000
29	0.400000	0.400000
30	0.400000	0.500000
31	0.000000	0.000000
32	0.000000	0.100000
33	0.000000	0.200000
34	0.000000	0.300000
35	0.000000	0.400000
36	0.000000	0.500000
37	2.00000	0.000000
38	2.00000	0.100000
39	2.00000	0.200000
40	2.00000	0.300000
41	2.00000	0.400000
42	2.00000	0.500000
43	2.11250	0.000000
44	2.11250	0.100000
45	2.11250	0.200000
46	2.11250	0.300000
47	2.11250	0.400000
48	2.11250	0.500000
49	2.22500	0.000000
50	2.22500	0.100000
51	2.22500	0.200000
52	2.22500	0.300000
53	2.22500	0.400000
54	2.22500	0.500000
55	2.33750	0.000000
56	2.33750	0.100000
57	2.33750	0.200000
58	2.33750	0.300000
59	2.33750	0.400000
60	2.33750	0.500000
61	2.45000	0.000000
62	2.45000	0.100000
63	2.45000	0.200000
64	2.45000	0.300000
65	2.45000	0.400000
66	2.45000	0.500000
67	2.56250	0.000000
68	2.56250	0.100000
69	2.56250	0.200000
70	2.56250	0.300000
71	2.56250	0.400000
72	2.56250	0.500000
73	2.67500	0.000000
74	2.67500	0.100000
75	2.67500	0.200000
76	2.67500	0.300000
77	2.67500	0.400000
78	2.67500	0.500000
79	2.78750	0.000000
80	2.78750	0.100000
81	2.78750	0.200000
82	2.78750	0.300000
83	2.78750	0.400000
84	2.78750	0.500000
85	2.90000	0.000000
86	2.90000	0.100000
87	2.90000	0.200000
88	2.90000	0.300000
89	2.90000	0.400000
90	2.90000	0.500000
91	3.01250	0.000000
92	3.01250	0.100000
93	3.01250	0.200000
94	3.01250	0.300000
95	3.01250	0.400000
96	3.01250	0.500000
97	3.12500	0.000000
98	3.12500	0.100000
99	3.12500	0.200000
100	3.12500	0.300000
101	3.12500	0.400000
102	3.12500	0.500000
103	3.23750	0.000000
104	3.23750	0.100000
105	3.23750	0.200000
106	3.23750	0.300000
107	3.23750	0.400000
108	3.23750	0.500000
109	3.35000	0.000000
110	3.35000	0.100000
111	3.35000	0.200000
112	3.35000	0.300000
113	3.35000	0.400000
114	3.35000	0.500000
115	3.46250	0.000000
116	3.46250	0.100000
117	3.46250	0.200000
118	3.46250	0.300000
119	3.46250	0.400000
120	3.46250	0.500000
121	3.57500	0.000000
122	3.57500	0.100000
123	3.57500	0.200000
124	3.57500	0.300000
125	3.57500	0.400000
126	3.57500	0.500000
127	3.68750	0.000000
128	3.68750	0.100000
129	3.68750	0.200000
130	3.68750	0.300000
131	3.68750	0.400000
132	3.68750	0.500000
133	3.80000	0.000000
134	3.80000	0.100000
135	3.80000	0.200000
136	3.80000	0.300000
137	3.80000	0.400000
138	3.80000	0.500000
139	3.91250	0.000000
140	3.91250	0.100000
141	3.91250	0.200000
142	3.91250	0.300000
143	3.91250	0.400000
144	3.91250	0.500000
145	4.02500	0.000000
146	4.02500	0.100000
147	4.02500	0.200000
148	4.02500	0.300000
149	4.02500	0.400000
150	4.02500	0.500000
151	4.13750	0.000000
152	4.13750	0.100000
153	4.13750	0.200000
154	4.13750	0.300000
155	4.13750	0.400000
156	4.13750	0.500000
157	4.25000	0.000000
158	4.25000	0.100000
159	4.25000	0.200000
160	4.25000	0.300000
161	4.25000	0.400000
162	4.25000	0.500000
163	4.25000	0.000000
164	4.32500	0.000000
165	4.40000	0.000000
166	4.47500	0.000000
167	4.55000	0.000000
168	4.62500	0.000000
169	4.70000	0.000000
170	4.77500	0.000000
171	4.85000	0.000000
172	4.92500	0.000000
173	5.00000	0.000000
174	5.07500	0.000000
175	5.15000	0.000000
176	5.22500	0.000000
177	5.30000	0.000000
178	5.37500	0.000000
179	5.45000	0.000000
180	5.52500	0.000000
181	5.60000	0.000000
182	5.67500	0.000000
183	5.75000	0.000000
184	4.25000	0.100000
185	4.32500	0.100000
186	4.40000	0.100000

187	4.47500	0.100000
188	4.55000	0.100000
189	4.62500	0.100000
190	4.70000	0.100000
191	4.77500	0.100000
192	4.85000	0.100000
193	4.92500	0.100000
194	5.00000	0.100000
195	5.07500	0.100000
196	5.15000	0.100000
197	5.22500	0.100000
198	5.30000	0.100000
199	5.37500	0.100000
200	5.45000	0.100000
201	5.52500	0.100000
202	5.60000	0.100000
203	5.67500	0.100000
204	5.75000	0.100000
205	4.25000	0.200000
206	4.32500	0.200000
207	4.40000	0.200000
208	4.47500	0.200000
209	4.55000	0.200000
210	4.62500	0.200000
211	4.70000	0.200000
212	4.77500	0.200000
213	4.85000	0.200000
214	4.92500	0.200000
215	5.00000	0.200000
216	5.07500	0.200000
217	5.15000	0.200000
218	5.22500	0.200000
219	5.30000	0.200000
220	5.37500	0.200000
221	5.45000	0.200000
222	5.52500	0.200000
223	5.60000	0.200000
224	5.67500	0.200000
225	5.75000	0.200000
226	4.25000	0.300000
227	4.32500	0.300000
228	4.40000	0.300000
229	4.47500	0.300000
230	4.55000	0.300000
231	4.62500	0.300000
232	4.70000	0.300000
233	4.77500	0.300000
234	4.85000	0.300000
235	4.92500	0.300000
236	5.00000	0.300000
237	5.07500	0.300000
238	5.15000	0.300000
239	5.22500	0.300000
240	5.30000	0.300000
241	5.37500	0.300000
242	5.45000	0.300000
243	5.52500	0.300000
244	5.60000	0.300000
245	5.67500	0.300000
246	5.75000	0.300000
247	4.25000	0.400000
248	4.32500	0.400000
249	4.40000	0.400000
250	4.47500	0.400000
251	4.55000	0.400000
252	4.62500	0.400000
253	4.70000	0.400000
254	4.77500	0.400000
255	4.85000	0.400000
256	4.92500	0.400000
257	5.00000	0.400000
258	5.07500	0.400000
259	5.15000	0.400000
260	5.22500	0.400000
261	5.30000	0.400000
262	5.37500	0.400000
263	5.45000	0.400000
264	5.52500	0.400000
265	5.60000	0.400000
266	5.67500	0.400000
267	5.75000	0.400000
268	4.25000	0.500000
269	4.32500	0.500000
270	4.40000	0.500000
271	4.47500	0.500000
272	4.55000	0.500000
273	4.62500	0.500000
274	4.70000	0.500000
275	4.77500	0.500000
276	4.85000	0.500000
277	4.92500	0.500000
278	5.00000	0.500000
279	5.07500	0.500000

280	5.15000	0.500000
281	5.22500	0.500000
282	5.30000	0.500000
283	5.37500	0.500000
284	5.45000	0.500000
285	5.52500	0.500000
286	5.60000	0.500000
287	5.67500	0.500000
288	5.75000	0.500000
289	5.75000	0.000000
290	5.86250	0.000000
291	5.97500	0.000000
292	6.08750	0.000000
293	6.20000	0.000000
294	6.31250	0.000000
295	6.42500	0.000000
296	6.53750	0.000000
297	6.65000	0.000000
298	6.76250	0.000000
299	6.87500	0.000000
300	6.98750	0.000000
301	7.10000	0.000000
302	7.21250	0.000000
303	7.32500	0.000000
304	7.43750	0.000000
305	7.55000	0.000000
306	7.66250	0.000000
307	7.77500	0.000000
308	7.88750	0.000000
309	8.00000	0.000000
310	5.75000	0.100000
311	5.86250	0.100000
312	5.97500	0.100000
313	6.08750	0.100000
314	6.20000	0.100000
315	6.31250	0.100000
316	6.42500	0.100000
317	6.53750	0.100000
318	6.65000	0.100000
319	6.76250	0.100000
320	6.87500	0.100000
321	6.98750	0.100000
322	7.10000	0.100000
323	7.21250	0.100000
324	7.32500	0.100000
325	7.43750	0.100000
326	7.55000	0.100000
327	7.66250	0.100000
328	7.77500	0.100000
329	7.88750	0.100000
330	8.00000	0.100000
331	5.75000	0.200000
332	5.86250	0.200000
333	5.97500	0.200000
334	6.08750	0.200000
335	6.20000	0.200000
336	6.31250	0.200000
337	6.42500	0.200000
338	6.53750	0.200000
339	6.65000	0.200000
340	6.76250	0.200000
341	6.87500	0.200000
342	6.98750	0.200000
343	7.10000	0.200000
344	7.21250	0.200000
345	7.32500	0.200000
346	7.43750	0.200000
347	7.55000	0.200000
348	7.66250	0.200000
349	7.77500	0.200000
350	7.88750	0.200000
351	8.00000	0.200000
352	5.75000	0.300000
353	5.86250	0.300000
354	5.97500	0.300000
355	6.08750	0.300000
356	6.20000	0.300000
357	6.31250	0.300000
358	6.42500	0.300000
359	6.53750	0.300000
360	6.65000	0.300000
361	6.76250	0.300000
362	6.87500	0.300000
363	6.98750	0.300000
364	7.10000	0.300000
365	7.21250	0.300000
366	7.32500	0.300000
367	7.43750	0.300000
368	7.55000	0.300000
369	7.66250	0.300000
370	7.77500	0.300000
371	7.88750	0.300000
372	8.00000	0.300000

373	5.75000	0.400000
374	5.86250	0.400000
375	5.97500	0.400000
376	6.08750	0.400000
377	6.20000	0.400000
378	6.31250	0.400000
379	6.42500	0.400000
380	6.53750	0.400000
381	6.65000	0.400000
382	6.76250	0.400000
383	6.87500	0.400000
384	6.98750	0.400000
385	7.10000	0.400000
386	7.21250	0.400000
387	7.32500	0.400000
388	7.43750	0.400000
389	7.55000	0.400000
390	7.66250	0.400000
391	7.77500	0.400000
392	7.88750	0.400000
393	8.00000	0.400000
394	5.75000	0.500000
395	5.86250	0.500000
396	5.97500	0.500000
397	6.08750	0.500000
398	6.20000	0.500000
399	6.31250	0.500000
400	6.42500	0.500000
401	6.53750	0.500000
402	6.65000	0.500000
403	6.76250	0.500000
404	6.87500	0.500000
405	6.98750	0.500000
406	7.10000	0.500000
407	7.21250	0.500000
408	7.32500	0.500000
409	7.43750	0.500000
410	7.55000	0.500000
411	7.66250	0.500000
412	7.77500	0.500000
413	7.88750	0.500000
414	8.00000	0.500000
415	8.00000	0.000000
416	8.06250	0.000000
417	8.12500	0.000000
418	8.18750	0.000000
419	8.25000	0.000000
420	8.31250	0.000000
421	8.37500	0.000000
422	8.43750	0.000000
423	8.50000	0.000000
424	8.00000	0.100000
425	8.06250	0.100000
426	8.12500	0.100000
427	8.18750	0.100000
428	8.25000	0.100000
429	8.31250	0.100000
430	8.37500	0.100000
431	8.43750	0.100000
432	8.50000	0.100000
433	8.00000	0.200000
434	8.06250	0.200000
435	8.12500	0.200000
436	8.18750	0.200000
437	8.25000	0.200000
438	8.31250	0.200000
439	8.37500	0.200000
440	8.43750	0.200000
441	8.50000	0.200000
442	8.00000	0.300000
443	8.06250	0.300000
444	8.12500	0.300000
445	8.18750	0.300000
446	8.25000	0.300000
447	8.31250	0.300000
448	8.37500	0.300000
449	8.43750	0.300000
450	8.50000	0.300000
451	8.00000	0.400000
452	8.06250	0.400000
453	8.12500	0.400000
454	8.18750	0.400000
455	8.25000	0.400000
456	8.31250	0.400000
457	8.37500	0.400000
458	8.43750	0.400000
459	8.50000	0.400000
460	8.00000	0.500000
461	8.06250	0.500000
462	8.12500	0.500000
463	8.18750	0.500000
464	8.25000	0.500000
465	8.31250	0.500000
466	8.37500	0.500000
467	8.43750	0.500000
468	8.50000	0.500000
469	8.50000	0.000000
470	9.00000	0.000000
471	9.50000	0.000000
472	10.0000	0.000000
473	10.5000	0.000000
474	8.50000	0.100000
475	9.00000	0.100000
476	9.50000	0.100000
477	10.0000	0.100000
478	10.5000	0.100000
479	8.50000	0.200000
480	9.00000	0.200000
481	9.50000	0.200000
482	10.0000	0.200000
483	10.5000	0.200000
484	8.50000	0.300000
485	9.00000	0.300000
486	9.50000	0.300000
487	10.0000	0.300000
488	10.5000	0.300000
489	8.50000	0.400000
490	9.00000	0.400000
491	9.50000	0.400000
492	10.0000	0.400000
493	10.5000	0.400000
494	8.50000	0.500000
495	9.00000	0.500000
496	9.50000	0.500000
497	10.0000	0.500000
498	10.5000	0.500000
499	2.00000	0.500000
500	2.11250	0.500000
501	2.22500	0.500000
502	2.33750	0.500000
503	2.45000	0.500000
504	2.56250	0.500000
505	2.67500	0.500000
506	2.78750	0.500000
507	2.90000	0.500000
508	3.01250	0.500000
509	3.12500	0.500000
510	3.23750	0.500000
511	3.35000	0.500000
512	3.46250	0.500000
513	3.57500	0.500000
514	3.68750	0.500000
515	3.80000	0.500000
516	3.91250	0.500000
517	4.02500	0.500000
518	4.13750	0.500000
519	4.25000	0.500000
520	2.00000	0.530000
521	2.11250	0.533500
522	2.22500	0.537000
523	2.33750	0.540500
524	2.45000	0.544000
525	2.56250	0.547500
526	2.67500	0.551000
527	2.78750	0.554500
528	2.90000	0.558000
529	3.01250	0.561500
530	3.12500	0.565000
531	3.23750	0.568500
532	3.35000	0.572000
533	3.46250	0.575500
534	3.57500	0.579000
535	3.68750	0.582500
536	3.80000	0.586000
537	3.91250	0.589500
538	4.02500	0.593000
539	4.13750	0.596500
540	4.25000	0.600000
541	2.00000	0.560000
542	2.11250	0.567000
543	2.22500	0.574000
544	2.33750	0.581000
545	2.45000	0.588000
546	2.56250	0.595000
547	2.67500	0.602000
548	2.78750	0.609000
549	2.90000	0.616000
550	3.01250	0.623000
551	3.12500	0.630000
552	3.23750	0.637000
553	3.35000	0.644000
554	3.46250	0.651000
555	3.57500	0.658000
556	3.68750	0.665000
557	3.80000	0.672000
558	3.91250	0.679000

559	4.02500	0.686000
560	4.13750	0.693000
561	4.25000	0.700000
562	2.00000	0.590000
563	2.11250	0.600500
564	2.22500	0.611000
565	2.33750	0.621500
566	2.45000	0.632000
567	2.56250	0.642500
568	2.67500	0.653000
569	2.78750	0.663500
570	2.90000	0.674000
571	3.01250	0.684500
572	3.12500	0.695000
573	3.23750	0.705500
574	3.35000	0.716000
575	3.46250	0.726500
576	3.57500	0.737000
577	3.68750	0.747500
578	3.80000	0.758000
579	3.91250	0.768500
580	4.02500	0.779000
581	4.13750	0.789500
582	4.25000	0.800000
583	2.00000	0.620000
584	2.11250	0.634000
585	2.22500	0.648000
586	2.33750	0.662000
587	2.45000	0.676000
588	2.56250	0.690000
589	2.67500	0.704000
590	2.78750	0.718000
591	2.90000	0.732000
592	3.01250	0.746000
593	3.12500	0.760000
594	3.23750	0.774000
595	3.35000	0.788000
596	3.46250	0.802000
597	3.57500	0.816000
598	3.68750	0.830000
599	3.80000	0.844000
600	3.91250	0.858000
601	4.02500	0.872000
602	4.13750	0.886000
603	4.25000	0.900000
604	2.00000	0.650000
605	2.11250	0.667500
606	2.22500	0.685000
607	2.33750	0.702500
608	2.45000	0.720000
609	2.56250	0.737500
610	2.67500	0.755000
611	2.78750	0.772500
612	2.90000	0.790000
613	3.01250	0.807500
614	3.12500	0.825000
615	3.23750	0.842500
616	3.35000	0.860000
617	3.46250	0.877500
618	3.57500	0.895000
619	3.68750	0.912500
620	3.80000	0.930000
621	3.91250	0.947500
622	4.02500	0.965000
623	4.13750	0.982500
624	4.25000	1.00000
625	4.25000	0.500000
626	4.32500	0.500000
627	4.40000	0.500000
628	4.47500	0.500000
629	4.55000	0.500000
630	4.62500	0.500000
631	4.70000	0.500000
632	4.77500	0.500000
633	4.85000	0.500000
634	4.92500	0.500000
635	5.00000	0.500000
636	5.07500	0.500000
637	5.15000	0.500000
638	5.22500	0.500000
639	5.30000	0.500000
640	5.37500	0.500000
641	5.45000	0.500000
642	5.52500	0.500000
643	5.60000	0.500000
644	5.67500	0.500000
645	5.75000	0.500000
646	4.25000	0.600000
647	4.32500	0.600000
648	4.40000	0.600000
649	4.47500	0.600000
650	4.55000	0.600000
651	4.62500	0.600000
652	4.70000	0.600000
653	4.77500	0.600000
654	4.85000	0.600000
655	4.92500	0.600000
656	5.00000	0.600000
657	5.07500	0.600000
658	5.15000	0.600000
659	5.22500	0.600000
660	5.30000	0.600000
661	5.37500	0.600000
662	5.45000	0.600000
663	5.52500	0.600000
664	5.60000	0.600000
665	5.67500	0.600000
666	5.75000	0.600000
667	4.25000	0.700000
668	4.32500	0.700000
669	4.40000	0.700000
670	4.47500	0.700000
671	4.55000	0.700000
672	4.62500	0.700000
673	4.70000	0.700000
674	4.77500	0.700000
675	4.85000	0.700000
676	4.92500	0.700000
677	5.00000	0.700000
678	5.07500	0.700000
679	5.15000	0.700000
680	5.22500	0.700000
681	5.30000	0.700000
682	5.37500	0.700000
683	5.45000	0.700000
684	5.52500	0.700000
685	5.60000	0.700000
686	5.67500	0.700000
687	5.75000	0.700000
688	4.25000	0.800000
689	4.32500	0.800000
690	4.40000	0.800000
691	4.47500	0.800000
692	4.55000	0.800000
693	4.62500	0.800000
694	4.70000	0.800000
695	4.77500	0.800000
696	4.85000	0.800000
697	4.92500	0.800000
698	5.00000	0.800000
699	5.07500	0.800000
700	5.15000	0.800000
701	5.22500	0.800000
702	5.30000	0.800000
703	5.37500	0.800000
704	5.45000	0.800000
705	5.52500	0.800000
706	5.60000	0.800000
707	5.67500	0.800000
708	5.75000	0.800000
709	4.25000	0.900000
710	4.32500	0.900000
711	4.40000	0.900000
712	4.47500	0.900000
713	4.55000	0.900000
714	4.62500	0.900000
715	4.70000	0.900000
716	4.77500	0.900000
717	4.85000	0.900000
718	4.92500	0.900000
719	5.00000	0.900000
720	5.07500	0.900000
721	5.15000	0.900000
722	5.22500	0.900000
723	5.30000	0.900000
724	5.37500	0.900000
725	5.45000	0.900000
726	5.52500	0.900000
727	5.60000	0.900000
728	5.67500	0.900000
729	5.75000	0.900000
730	4.25000	0.999999
731	4.32500	0.999999
732	4.40000	0.999999
733	4.47500	0.999999
734	4.55000	0.999999
735	4.62500	0.999999
736	4.70000	1.00000
737	4.77500	1.00000
738	4.85000	1.00000
739	4.92500	1.00000
740	5.00000	1.00000
741	5.07500	1.00000
742	5.15000	1.00000
743	5.22500	1.00000
744	5.30000	1.00000

745	5.37500	1.00000
746	5.45000	1.00000
747	5.52500	1.00000
748	5.60000	1.00000
749	5.67500	1.00000
750	5.75000	1.00000
751	4.32854	1.04438
752	4.39569	1.04438
753	4.46283	1.04438
754	4.52998	1.04438
755	4.59713	1.04438
756	4.66427	1.04438
757	4.73142	1.04438
758	4.79856	1.04438
759	4.86571	1.04438
760	4.93285	1.04438
761	5.00000	1.04438
762	5.06714	1.04439
763	5.13429	1.04439
764	5.20144	1.04439
765	5.26858	1.04439
766	5.33573	1.04439
767	5.40287	1.04439
768	5.47002	1.04439
769	5.53716	1.04439
770	5.60431	1.04439
771	5.67145	1.04439
772	4.39783	1.10216
773	4.45805	1.10216
774	4.51827	1.10216
775	4.57848	1.10216
776	4.63870	1.10216
777	4.69892	1.10216
778	4.75913	1.10216
779	4.81935	1.10216
780	4.87956	1.10216
781	4.93978	1.10216
782	5.00000	1.10216
783	5.06021	1.10216
784	5.12043	1.10216
785	5.18065	1.10216
786	5.24086	1.10216
787	5.30108	1.10216
788	5.36130	1.10216
789	5.42151	1.10216
790	5.48173	1.10216
791	5.54195	1.10216
792	5.60216	1.10216
793	4.45561	1.17145
794	4.51005	1.17145
795	4.56449	1.17145
796	4.61893	1.17145
797	4.67337	1.17145
798	4.72781	1.17145
799	4.78225	1.17145
800	4.83668	1.17145
801	4.89112	1.17145
802	4.94556	1.17145
803	5.00000	1.17145
804	5.05444	1.17145
805	5.10888	1.17145
806	5.16331	1.17146
807	5.21775	1.17146
808	5.27219	1.17146
809	5.32663	1.17146
810	5.38107	1.17146
811	5.43551	1.17146
812	5.48995	1.17146
813	5.54438	1.17146
814	4.50000	1.25000
815	4.55000	1.25000
816	4.60000	1.25000
817	4.65000	1.25000
818	4.70000	1.25000
819	4.75000	1.25000
820	4.80000	1.25000
821	4.85000	1.25000
822	4.90000	1.25000
823	4.95000	1.25000
824	5.00000	1.25000
825	5.05000	1.25000
826	5.10000	1.25000
827	5.15000	1.25000
828	5.20000	1.25000
829	5.25000	1.25000
830	5.30000	1.25000
831	5.35000	1.25000
832	5.40000	1.25000
833	5.45000	1.25000
834	5.50000	1.25000
835	4.50000	1.33333
836	4.55000	1.33333
837	4.60000	1.33333
838	4.65000	1.33333
839	4.70000	1.33333
840	4.75000	1.33333
841	4.80000	1.33333
842	4.85000	1.33333
843	4.90000	1.33333
844	4.95000	1.33333
845	5.00000	1.33333
846	5.05000	1.33333
847	5.10000	1.33333
848	5.15000	1.33333
849	5.20000	1.33333
850	5.25000	1.33333
851	5.30000	1.33333
852	5.35000	1.33333
853	5.40000	1.33333
854	5.45000	1.33333
855	5.50000	1.33333
856	4.50000	1.41667
857	4.55000	1.41667
858	4.60000	1.41667
859	4.65000	1.41667
860	4.70000	1.41667
861	4.75000	1.41667
862	4.80000	1.41667
863	4.85000	1.41667
864	4.90000	1.41667
865	4.95000	1.41667
866	5.00000	1.41667
867	5.05000	1.41667
868	5.10000	1.41667
869	5.15000	1.41667
870	5.20000	1.41667
871	5.25000	1.41667
872	5.30000	1.41667
873	5.35000	1.41667
874	5.40000	1.41667
875	5.45000	1.41667
876	5.50000	1.41667
877	4.50000	1.50000
878	4.55000	1.50000
879	4.60000	1.50000
880	4.65000	1.50000
881	4.70000	1.50000
882	4.75000	1.50000
883	4.80000	1.50000
884	4.85000	1.50000
885	4.90000	1.50000
886	4.95000	1.50000
887	5.00000	1.50000
888	5.05000	1.50000
889	5.10000	1.50000
890	5.15000	1.50000
891	5.20000	1.50000
892	5.25000	1.50000
893	5.30000	1.50000
894	5.35000	1.50000
895	5.40000	1.50000
896	5.45000	1.50000
897	5.50000	1.50000
898	5.75000	0.500000
899	5.86250	0.500000
900	5.97500	0.500000
901	6.08750	0.500000
902	6.20000	0.500000
903	6.31250	0.500000
904	6.42500	0.500000
905	6.53750	0.500000
906	6.65000	0.500000
907	6.76250	0.500000
908	6.87500	0.500000
909	6.98750	0.500000
910	7.10000	0.500000
911	7.21250	0.500000
912	7.32500	0.500000
913	7.43750	0.500000
914	7.55000	0.500000
915	7.66250	0.500000
916	7.77500	0.500000
917	7.88750	0.500000
918	8.00000	0.500000
919	5.75000	0.600000
920	5.86250	0.597000
921	5.97500	0.594000
922	6.08750	0.591000
923	6.20000	0.588000
924	6.31250	0.585000
925	6.42500	0.582000
926	6.53750	0.579000
927	6.65000	0.576000
928	6.76250	0.573000
929	6.87500	0.570000
930	6.98750	0.567000

931	7.10000	0.564000
932	7.21250	0.561000
933	7.32500	0.558000
934	7.43750	0.555000
935	7.55000	0.552000
936	7.66250	0.549000
937	7.77500	0.546000
938	7.88750	0.543000
939	8.00000	0.540000
940	5.75000	0.700000
941	5.86250	0.694000
942	5.97500	0.688000
943	6.08750	0.682000
944	6.20000	0.676000
945	6.31250	0.670000
946	6.42500	0.664000
947	6.53750	0.658000
948	6.65000	0.652000
949	6.76250	0.646000
950	6.87500	0.640000
951	6.98750	0.634000
952	7.10000	0.628000
953	7.21250	0.622000
954	7.32500	0.616000
955	7.43750	0.610000
956	7.55000	0.604000
957	7.66250	0.598000
958	7.77500	0.592000
959	7.88750	0.586000
960	8.00000	0.580000
961	5.75000	0.800000
962	5.86250	0.791000
963	5.97500	0.782000
964	6.08750	0.773000
965	6.20000	0.764000
966	6.31250	0.755000
967	6.42500	0.746000
968	6.53750	0.737000
969	6.65000	0.728000
970	6.76250	0.719000
971	6.87500	0.710000
972	6.98750	0.701000
973	7.10000	0.692000
974	7.21250	0.683000
975	7.32500	0.674000
976	7.43750	0.665000
977	7.55000	0.656000
978	7.66250	0.647000
979	7.77500	0.638000
980	7.88750	0.629000
981	8.00000	0.620000
982	5.75000	0.900000
983	5.86250	0.888000
984	5.97500	0.876000
985	6.08750	0.864000
986	6.20000	0.852000
987	6.31250	0.840000
988	6.42500	0.828000
989	6.53750	0.816000
990	6.65000	0.804000
991	6.76250	0.792000
992	6.87500	0.780000
993	6.98750	0.768000
994	7.10000	0.756000
995	7.21250	0.744000
996	7.32500	0.732000
997	7.43750	0.720000
998	7.55000	0.708000
999	7.66250	0.696000
1000	7.77500	0.684000
1001	7.88750	0.672000
1002	8.00000	0.660000
1003	5.75000	1.00000
1004	5.86250	0.985000
1005	5.97500	0.970000
1006	6.08750	0.955000
1007	6.20000	0.940000
1008	6.31250	0.925000
1009	6.42500	0.910000
1010	6.53750	0.895000
1011	6.65000	0.880000
1012	6.76250	0.865000
1013	6.87500	0.850000
1014	6.98750	0.835000
1015	7.10000	0.820000
1016	7.21250	0.805000
1017	7.32500	0.790000
1018	7.43750	0.775000
1019	7.55000	0.760000
1020	7.66250	0.745000
1021	7.77500	0.730000
1022	7.88750	0.715000
1023	8.00000	0.700000
1024	8.00000	0.700000
1025	8.10364	0.705385
1026	8.20617	0.721483
1027	8.30647	0.748120
1028	8.40348	0.785010
1029	8.50000	0.825000
1030	8.58333	0.800000
1031	8.66667	0.775000
1032	8.75000	0.750000
1033	8.00000	0.660000
1034	8.08581	0.664308
1035	8.17163	0.677186
1036	8.25744	0.698496
1037	8.34325	0.728008
1038	8.42906	0.760000
1039	8.51488	0.740000
1040	8.60069	0.720000
1041	8.68650	0.715616
1042	8.00000	0.620000
1043	8.07858	0.623231
1044	8.15715	0.632890
1045	8.23573	0.648872
1046	8.31431	0.671006
1047	8.39289	0.695000
1048	8.47146	0.680000
1049	8.55004	0.665000
1050	8.62862	0.672444
1051	8.00000	0.580000
1052	8.07220	0.582154
1053	8.14439	0.588593
1054	8.21658	0.599248
1055	8.28878	0.614004
1056	8.36097	0.630000
1057	8.43317	0.620000
1058	8.50536	0.610000
1059	8.57756	0.621383
1060	8.00000	0.540000
1061	8.06680	0.541077
1062	8.13360	0.544297
1063	8.20039	0.549624
1064	8.26719	0.557002
1065	8.33399	0.565000
1066	8.40079	0.560000
1067	8.46759	0.555000
1068	8.53438	0.563500
1069	8.00000	0.500000
1070	8.06250	0.500000
1071	8.12500	0.500000
1072	8.18750	0.500000
1073	8.25000	0.500000
1074	8.31250	0.500000
1075	8.37500	0.500000
1076	8.43750	0.500000
1077	8.50000	0.500000
1078	10.0000	1.35000
1079	10.0000	1.40000
1080	10.0000	1.45000
1081	10.0000	1.50000
1082	9.86667	1.35000
1083	9.85556	1.40000
1084	9.84444	1.45000
1085	9.83333	1.50000
1086	9.73333	1.35000
1087	9.71111	1.40000
1088	9.68889	1.45000
1089	9.66667	1.50000
1090	9.60000	1.35000
1091	9.56667	1.40000
1092	9.53333	1.45000
1093	9.50000	1.50000
1094	9.51500	1.29000
1095	9.47667	1.33750
1096	9.43833	1.38500
1097	9.40000	1.43250
1098	9.43000	1.23000
1099	9.38667	1.27500
1100	9.34333	1.32000
1101	9.30000	1.36500
1102	9.34500	1.17000
1103	9.29667	1.21250
1104	9.24833	1.25500
1105	9.20000	1.29750
1106	9.26000	1.11000
1107	9.20667	1.15000
1108	9.15333	1.19000
1109	9.10000	1.23000
1110	9.17500	1.05000
1111	9.11667	1.08750
1112	9.05833	1.12500
1113	9.00000	1.16250
1114	9.09000	0.990000
1115	9.02667	1.02500
1116	8.96333	1.06000
1117	8.90000	1.09500
1118	9.00500	0.930000
1119	8.93667	0.962500
1120	8.86833	0.995000
1121	8.80000	1.02750
1122	8.92000	0.870000
1123	8.84667	0.900000
1124	8.77333	0.930000
1125	8.70000	0.960000
1126	8.83500	0.810000
1127	8.75667	0.837500
1128	8.67833	0.865000
1129	8.60000	0.892500
1130	8.75000	0.750000
1131	8.66667	0.775000
1132	8.58333	0.800000
1133	8.50000	0.825000

A.3-- ELEMENT FILE OUTPUT

1	1	2	8	7	1
2	2	3	9	8	1
3	3	4	10	9	1
4	4	5	11	10	1
5	5	6	12	11	1
6	7	8	14	13	1
7	8	9	15	14	1
8	9	10	16	15	1
9	10	11	17	16	1
10	11	12	18	17	1
11	13	14	20	19	1
12	14	15	21	20	1
13	15	16	22	21	1
14	16	17	23	22	1
15	17	18	24	23	1
16	19	20	26	25	1
17	20	21	27	26	1
18	21	22	28	27	1
19	22	23	29	28	1
20	23	24	30	29	1
21	25	26	32	31	1
22	26	27	33	32	1
23	27	28	34	33	1
24	28	29	35	34	1
25	29	30	36	35	1
26	37	38	44	43	1
27	38	39	45	44	1
28	39	40	46	45	1
29	40	41	47	46	1
30	41	42	48	47	1
31	43	44	50	49	1
32	44	45	51	50	1
33	45	46	52	51	1
34	46	47	53	52	1
35	47	48	54	53	1
36	49	50	56	55	1
37	50	51	57	56	1
38	51	52	58	57	1
39	52	53	59	58	1
40	53	54	60	59	1
41	55	56	62	61	1
42	56	57	63	62	1
43	57	58	64	63	1
44	58	59	65	64	1
45	59	60	66	65	1
46	61	62	68	67	1
47	62	63	69	68	1
48	63	64	70	69	1
49	64	65	71	70	1
50	65	66	72	71	1
51	67	68	74	73	1
52	68	69	75	74	1
53	69	70	76	75	1
54	70	71	77	76	1
55	71	72	78	77	1
56	73	74	80	79	1
57	74	75	81	80	1
58	75	76	82	81	1
59	76	77	83	82	1
60	77	78	84	83	1
61	79	80	86	85	1
62	80	81	87	86	1
63	81	82	88	87	1
64	82	83	89	88	1
65	83	84	90	89	1
66	85	86	92	91	1
67	86	87	93	92	1
68	87	88	94	93	1
69	88	89	95	94	1
70	89	90	96	95	1
71	91	92	98	97	1
72	92	93	99	98	1
73	93	94	100	99	1
74	94	95	101	100	1
75	95	96	102	101	1
76	97	98	104	103	1
77	98	99	105	104	1
78	99	100	106	105	1
79	100	101	107	106	1
80	101	102	108	107	1
81	103	104	110	109	1
82	104	105	111	110	1
83	105	106	112	111	1
84	106	107	113	112	1
85	107	108	114	113	1
86	109	110	116	115	1
87	110	111	117	116	1
88	111	112	118	117	1
89	112	113	119	118	1
90	113	114	120	119	1
91	115	116	122	121	1
92	116	117	123	122	1
93	117	118	124	123	1

94	118	119	125	124	1
95	119	120	126	125	1
96	121	122	128	127	1
97	122	123	129	128	1
98	123	124	130	129	1
99	124	125	131	130	1
100	125	126	132	131	1
101	127	128	134	133	1
102	128	129	135	134	1
103	129	130	136	135	1
104	130	131	137	136	1
105	131	132	138	137	1
106	133	134	140	139	1
107	134	135	141	140	1
108	135	136	142	141	1
109	136	137	143	142	1
110	137	138	144	143	1
111	139	140	146	145	1
112	140	141	147	146	1
113	141	142	148	147	1
114	142	143	149	148	1
115	143	144	150	149	1
116	145	146	152	151	1
117	146	147	153	152	1
118	147	148	154	153	1
119	148	149	155	154	1
120	149	150	156	155	1
121	151	152	158	157	1
122	152	153	159	158	1
123	153	154	160	159	1
124	154	155	161	160	1
125	155	156	162	161	1
126	163	164	185	184	1
127	164	165	186	185	1
128	165	166	187	186	1
129	166	167	188	187	1
130	167	168	189	188	1
131	168	169	190	189	1
132	169	170	191	190	1
133	170	171	192	191	1
134	171	172	193	192	1
135	172	173	194	193	1
136	173	174	195	194	1
137	174	175	196	195	1
138	175	176	197	196	1
139	176	177	198	197	1
140	177	178	199	198	1
141	178	179	200	199	1
142	179	180	201	200	1
143	180	181	202	201	1
144	181	182	203	202	1
145	182	183	204	203	1
146	184	185	206	205	1
147	185	186	207	206	1
148	186	187	208	207	1
149	187	188	209	208	1
150	188	189	210	209	1
151	189	190	211	210	1
152	190	191	212	211	1
153	191	192	213	212	1
154	192	193	214	213	1
155	193	194	215	214	1
156	194	195	216	215	1
157	195	196	217	216	1
158	196	197	218	217	1
159	197	198	219	218	1
160	198	199	220	219	1
161	199	200	221	220	1
162	200	201	222	221	1
163	201	202	223	222	1
164	202	203	224	223	1
165	203	204	225	224	1
166	205	206	227	226	1
167	206	207	228	227	1
168	207	208	229	228	1
169	208	209	230	229	1
170	209	210	231	230	1
171	210	211	232	231	1
172	211	212	233	232	1
173	212	213	234	233	1
174	213	214	235	234	1
175	214	215	236	235	1
176	215	216	237	236	1
177	216	217	238	237	1
178	217	218	239	238	1
179	218	219	240	239	1
180	219	220	241	240	1
181	220	221	242	241	1
182	221	222	243	242	1
183	222	223	244	243	1
184	223	224	245	244	1
185	224	225	246	245	1
186	226	227	248	247	1

187	227	228	249	248	1
188	228	229	250	249	1
189	229	230	251	250	1
190	230	231	252	251	1
191	231	232	253	252	1
192	232	233	254	253	1
193	233	234	255	254	1
194	234	235	256	255	1
195	235	236	257	256	1
196	236	237	258	257	1
197	237	238	259	258	1
198	238	239	260	259	1
199	239	240	261	260	1
200	240	241	262	261	1
201	241	242	263	262	1
202	242	243	264	263	1
203	243	244	265	264	1
204	244	245	266	265	1
205	245	246	267	266	1
206	247	248	269	268	1
207	248	249	270	269	1
208	249	250	271	270	1
209	250	251	272	271	1
210	251	252	273	272	1
211	252	253	274	273	1
212	253	254	275	274	1
213	254	255	276	275	1
214	255	256	277	276	1
215	256	257	278	277	1
216	257	258	279	278	1
217	258	259	280	279	1
218	259	260	281	280	1
219	260	261	282	281	1
220	261	262	283	282	1
221	262	263	284	283	1
222	263	264	285	284	1
223	264	265	286	285	1
224	265	266	287	286	1
225	266	267	288	287	1
226	289	290	311	310	1
227	290	291	312	311	1
228	291	292	313	312	1
229	292	293	314	313	1
230	293	294	315	314	1
231	294	295	316	315	1
232	295	296	317	316	1
233	296	297	318	317	1
234	297	298	319	318	1
235	298	299	320	319	1
236	299	300	321	320	1
237	300	301	322	321	1
238	301	302	323	322	1
239	302	[illegible]	324	323	1
240	303	304	325	324	1
241	304	305	326	325	1
242	305	306	327	326	1
243	306	307	328	327	1
244	307	308	329	328	1
245	308	309	330	329	1
246	310	311	332	331	1
247	311	312	333	332	1
248	312	313	334	333	1
249	313	314	335	334	1
250	314	315	336	335	1
251	315	316	337	336	1
252	316	317	338	337	1
253	317	318	339	338	1
254	318	319	340	339	1
255	319	320	341	340	1
256	320	321	342	341	1
257	321	322	343	342	1
258	322	323	344	343	1
259	323	324	345	344	1
260	324	325	346	345	1
261	325	326	347	346	1
262	326	327	348	347	1
263	327	328	349	348	1
264	328	329	350	349	1
265	329	330	351	350	1
266	331	332	353	352	1
267	332	333	354	353	1
268	333	334	355	354	1
269	334	335	356	355	1
270	335	336	357	356	1
271	336	337	358	357	1
272	337	338	359	358	1
273	338	339	360	359	1
274	339	340	361	360	1
275	340	341	362	361	1
276	341	342	363	362	1
277	342	343	364	363	1
278	343	344	365	364	1
279	344	345	366	365	1

280	345	346	367	366	1
281	346	347	368	367	1
282	347	348	369	368	1
283	348	349	370	369	1
284	349	350	371	370	1
285	350	351	372	371	1
286	352	353	374	373	1
287	353	354	375	374	1
288	354	355	376	375	1
289	355	356	377	376	1
290	356	357	378	377	1
291	357	358	379	378	1
292	358	359	380	379	1
293	359	360	381	380	1
294	360	361	382	381	1
295	361	362	383	382	1
296	362	363	384	383	1
297	363	364	385	384	1
298	364	365	386	385	1
299	365	366	387	386	1
300	366	367	388	387	1
301	367	368	389	388	1
302	368	369	390	389	1
303	369	370	391	390	1
304	370	371	392	391	1
305	371	372	393	392	1
306	373	374	395	394	1
307	374	375	396	395	1
308	375	376	397	396	1
309	376	377	398	397	1
310	377	378	399	398	1
311	378	379	400	399	1
312	379	380	401	400	1
313	380	381	402	401	1
314	381	382	403	402	1
315	382	383	404	403	1
316	383	384	405	404	1
317	384	385	406	405	1
318	385	386	407	406	1
319	386	387	408	407	1
320	387	388	409	408	1
321	388	389	410	409	1
322	389	390	411	410	1
323	390	391	412	411	1
324	391	392	413	412	1
325	392	393	414	413	1
326	415	416	425	424	1
327	416	417	426	425	1
328	417	418	427	426	1
329	418	419	428	427	1
330	419	420	429	428	1
331	420	421	430	429	1
332	421	422	431	430	1
333	422	423	432	431	1
334	424	425	434	433	1
335	425	426	435	434	1
336	426	427	436	435	1
337	427	428	437	436	1
338	428	429	438	437	1
339	429	430	439	438	1
340	430	431	440	439	1
341	431	432	441	440	1
342	433	434	443	442	1
343	434	435	444	443	1
344	435	436	445	444	1
345	436	437	446	445	1
346	437	438	447	446	1
347	438	439	448	447	1
348	439	440	449	448	1
349	440	441	450	449	1
350	442	443	452	451	1
351	443	444	453	452	1
352	444	445	454	453	1
353	445	446	455	454	1
354	446	447	456	455	1
355	447	448	457	456	1
356	448	449	458	457	1
357	449	450	459	458	1
358	451	452	461	460	1
359	452	453	462	461	1
360	453	454	463	462	1
361	454	455	464	463	1
362	455	456	465	464	1
363	456	457	466	465	1
364	457	458	467	466	1
365	458	459	468	467	1
366	469	470	475	474	1
367	470	471	476	475	1
368	471	472	477	476	1
369	472	473	478	477	1
370	474	475	480	479	1
371	475	476	481	480	1
372	476	477	482	481	1

373	477	478	483	482	1
374	479	480	485	484	1
375	480	481	486	485	1
376	481	482	487	486	1
377	482	483	488	487	1
378	484	485	490	489	1
379	485	486	491	490	1
380	486	487	492	491	1
381	487	488	493	492	1
382	489	490	495	494	1
383	490	491	496	495	1
384	491	492	497	496	1
385	492	493	498	497	1
386	499	500	521	520	2
387	500	501	522	521	2
388	501	502	523	522	2
389	502	503	524	523	2
390	503	504	525	524	2
391	504	505	526	525	2
392	505	506	527	526	2
393	506	507	528	527	2
394	507	508	529	528	2
395	508	509	530	529	2
396	509	510	531	530	2
397	510	511	532	531	2
398	511	512	533	532	2
399	512	513	534	533	2
400	513	514	535	534	2
401	514	515	536	535	2
402	515	516	537	536	2
403	516	517	538	537	2
404	517	518	539	538	2
405	518	519	540	539	2
406	520	521	542	541	2
407	521	522	543	542	2
408	522	523	544	543	2
409	523	524	545	544	2
410	524	525	546	545	2
411	525	526	547	546	2
412	526	527	548	547	2
413	527	528	549	548	2
414	528	529	550	549	2
415	529	530	551	550	2
416	530	531	552	551	2
417	531	532	553	552	2
418	532	533	554	553	2
419	533	534	555	554	2
420	534	535	556	555	2
421	535	536	557	556	2
422	536	537	558	557	2
423	537	538	559	558	2
424	538	539	560	559	2
425	539	540	561	560	2
426	541	542	563	562	2
427	542	543	564	563	2
428	543	544	565	564	2
429	544	545	566	565	2
430	545	546	567	566	2
431	546	547	568	567	2
432	547	548	569	568	2
433	548	549	570	569	2
434	549	550	571	570	2
435	550	551	572	571	2
436	551	552	573	572	2
437	552	553	574	573	2
438	553	554	575	574	2
439	554	555	576	575	2
440	555	556	577	576	2
441	556	557	578	577	2
442	557	558	579	578	2
443	558	559	580	579	2
444	559	560	581	580	2
445	560	561	582	581	2
446	562	563	584	583	2
447	563	564	585	584	2
448	564	565	586	585	2
449	565	566	587	586	2
450	566	567	588	587	2
451	567	568	589	588	2
452	568	569	590	589	2
453	569	570	591	590	2
454	570	571	592	591	2
455	571	572	593	592	2
456	572	573	594	593	2
457	573	574	595	594	2
458	574	575	596	595	2
459	575	576	597	596	2
460	576	577	598	597	2
461	577	578	599	598	2
462	578	579	600	599	2
463	579	580	601	600	2
464	580	581	602	601	2
465	581	582	603	602	2

466	583	584	605	604	2
467	584	585	606	605	2
468	585	586	607	606	2
469	586	587	608	607	2
470	587	588	609	608	2
471	588	589	610	609	2
472	589	590	611	610	2
473	590	591	612	611	2
474	591	592	613	612	2
475	592	593	614	613	2
476	593	594	615	614	2
477	594	595	616	615	2
478	595	596	617	616	2
479	596	597	618	617	2
480	597	598	619	618	2
481	598	599	620	619	2
482	599	600	621	620	2
483	600	601	622	621	2
484	601	602	623	622	2
485	602	603	624	623	2
486	625	626	647	646	2
487	626	627	648	647	2
488	627	628	649	648	2
489	628	629	650	649	2
490	629	630	651	650	2
491	630	631	652	651	2
492	631	632	653	652	2
493	632	633	654	653	2
494	633	634	655	654	2
495	634	635	656	655	2
496	635	636	657	656	2
497	636	637	658	657	2
498	637	638	659	658	2
499	638	639	660	659	2
500	639	640	661	660	2
501	640	641	662	661	2
502	641	642	663	662	2
503	642	643	664	663	2
504	643	644	665	664	2
505	644	645	666	665	2
506	646	647	668	667	2
507	647	648	669	668	2
508	648	649	670	669	2
509	649	650	671	670	2
510	650	651	672	671	2
511	651	652	673	672	2
512	652	653	674	673	2
513	653	654	675	674	2
514	654	655	676	675	2
515	655	656	677	676	2
516	656	657	678	677	2
517	657	658	679	678	2
518	658	659	680	679	2
519	659	660	681	680	2
520	660	661	682	681	2
521	661	662	683	682	2
522	662	663	684	683	2
523	663	664	685	684	2
524	664	665	686	685	2
525	665	666	687	686	2
526	667	668	689	688	2
527	668	669	690	689	2
528	669	670	691	690	2
529	670	671	692	691	2
530	671	672	693	692	2
531	672	673	694	693	2
532	673	674	695	694	2
533	674	675	696	695	2
534	675	676	697	696	2
535	676	677	698	697	2
536	677	678	699	698	2
537	678	679	700	699	2
538	679	680	701	700	2
539	680	681	702	701	2
540	681	682	703	702	2
541	682	683	704	703	2
542	683	684	705	704	2
543	684	685	706	705	2
544	685	686	707	706	2
545	686	687	708	707	2
546	688	689	710	709	2
547	689	690	711	710	2
548	690	691	712	711	2
549	691	692	713	712	2
550	692	693	714	713	2
551	693	694	715	714	2
552	694	695	716	715	2
553	695	696	717	716	2
554	696	697	718	717	2
555	697	698	719	718	2
556	698	699	720	719	2
557	699	700	721	720	2
558	700	701	722	721	2

559	701	702	723	722	2
560	702	703	724	723	2
561	703	704	725	724	2
562	704	705	726	725	2
563	705	706	727	726	2
564	706	707	728	727	2
565	707	708	729	728	2
566	709	710	731	730	2
567	710	711	732	731	2
568	711	712	733	732	2
569	712	713	734	733	2
570	713	714	735	734	2
571	714	715	736	735	2
572	715	716	737	736	2
573	716	717	738	737	2
574	717	718	739	738	2
575	718	719	740	739	2
576	719	720	741	740	2
577	720	721	742	741	2
578	721	722	743	742	2
579	722	723	744	743	2
580	723	724	745	744	2
581	724	725	746	745	2
582	725	726	747	746	2
583	726	727	748	747	2
584	727	728	749	748	2
585	728	729	750	749	2
586	730	731	752	751	2
587	731	732	753	752	2
588	732	733	754	753	2
589	733	734	755	754	2
590	734	735	756	755	2
591	735	736	757	756	2
592	736	737	758	757	2
593	737	738	759	758	2
594	738	739	760	759	2
595	739	740	761	760	2
596	740	741	762	761	2
597	741	742	763	762	2
598	742	743	764	763	2
599	743	744	765	764	2
600	744	745	766	765	2
601	745	746	767	766	2
602	746	747	768	767	2
603	747	748	769	768	2
604	748	749	770	769	2
605	749	750	771	770	2
606	751	752	773	772	2
607	752	753	774	773	2
608	753	754	775	774	2
609	754	755	776	775	2
610	755	756	777	776	2
611	756	757	778	777	2
612	757	758	779	778	2
613	758	759	780	779	2
614	759	760	781	780	2
615	760	761	782	781	2
616	761	762	783	782	2
617	762	763	784	783	2
618	763	764	785	784	2
619	764	765	786	785	2
620	765	766	787	786	2
621	766	767	788	787	2
622	767	768	789	788	2
623	768	769	790	789	2
624	769	770	791	790	2
625	770	771	792	791	2
626	772	773	794	793	2
627	773	774	795	794	2
628	774	775	796	795	2
629	775	776	797	796	2
630	776	777	798	797	2
631	777	778	799	798	2
632	778	779	800	799	2
633	779	780	801	800	2
634	780	781	802	801	2
635	781	782	803	802	2
636	782	783	804	803	2
637	783	784	805	804	2
638	784	785	806	805	2
639	785	786	807	806	2
640	786	787	808	807	2
641	787	788	809	808	2
642	788	789	810	809	2
643	789	790	811	810	2
644	790	791	812	811	2
645	791	792	813	812	2
646	793	794	815	814	2
647	794	795	816	815	2
648	795	796	817	816	2
649	796	797	818	817	2
650	797	798	819	818	2
651	798	799	820	819	2

652	799	800	821	820	2
653	800	801	822	821	2
654	801	802	823	822	2
655	802	803	824	823	2
656	803	804	825	824	2
657	804	805	826	825	2
658	805	806	827	826	2
659	806	807	828	827	2
660	807	808	829	828	2
661	808	809	830	829	2
662	809	810	831	830	2
663	810	811	832	831	2
664	811	812	833	832	2
665	812	813	834	833	2
666	814	815	836	835	2
667	815	816	837	836	2
668	816	817	838	837	2
669	817	818	839	838	2
670	818	819	840	839	2
671	819	820	841	840	2
672	820	821	842	841	2
673	821	822	843	842	2
674	822	823	844	843	2
675	823	824	845	844	2
676	824	825	846	845	2
677	825	826	847	846	2
678	826	827	848	847	2
679	827	828	849	848	2
680	828	829	850	849	2
681	829	830	851	850	2
682	830	831	852	851	2
683	831	832	853	852	2
684	832	833	854	853	2
685	833	834	855	854	2
686	835	836	857	856	2
687	836	837	858	857	2
688	837	838	859	858	2
689	838	839	860	859	2
690	839	840	861	860	2
691	840	841	862	861	2
692	841	842	863	862	2
693	842	843	864	863	2
694	843	844	865	864	2
695	844	845	866	865	2
696	845	846	867	866	2
697	846	847	868	867	2
698	847	848	869	868	2
699	848	849	870	869	2
700	849	850	871	870	2
701	850	851	872	871	2
702	851	852	873	872	2
703	852	853	874	873	2
704	853	854	875	874	2
705	854	855	876	875	2
706	856	857	878	877	2
707	857	858	879	878	2
708	858	859	880	879	2
709	859	860	881	880	2
710	860	861	882	881	2
711	861	862	883	882	2
712	862	863	884	883	2
713	863	864	885	884	2
714	864	865	886	885	2
715	865	866	887	886	2
716	866	867	888	887	2
717	867	868	889	888	2
718	868	869	890	889	2
719	869	870	891	890	2
720	870	871	892	891	2
721	871	872	893	892	2
722	872	873	894	893	2
723	873	874	895	894	2
724	874	875	896	895	2
725	875	876	897	896	2
726	898	899	920	919	2
727	899	900	921	920	2
728	900	901	922	921	2
729	901	902	923	922	2
730	902	903	924	923	2
731	903	904	925	924	2
732	904	905	926	925	2
733	905	906	927	926	2
734	906	907	928	927	2
735	907	908	929	928	2
736	908	909	930	929	2
737	909	910	931	930	2
738	910	911	932	931	2
739	911	912	933	932	2
740	912	913	934	933	2
741	913	914	935	934	2
742	914	915	936	935	2
743	915	916	937	936	2
744	916	917	938	937	2

745	917	918	939	938	2
746	919	920	941	940	2
747	920	921	942	941	2
748	921	922	943	942	2
749	922	923	944	943	2
750	923	924	945	944	2
751	924	925	946	945	2
752	925	926	947	946	2
753	926	927	948	947	2
754	927	928	949	948	2
755	928	929	950	949	2
756	929	930	951	950	2
757	930	931	952	951	2
758	931	932	953	952	2
759	932	933	954	953	2
760	933	934	955	954	2
761	934	935	956	955	2
762	935	936	957	956	2
763	936	937	958	957	2
764	937	938	959	958	2
765	938	939	960	959	2
766	940	941	962	961	2
767	941	942	963	962	2
768	942	943	964	963	2
769	943	944	965	964	2
770	944	945	966	965	2
771	945	946	967	966	2
772	946	947	968	967	2
773	947	948	969	968	2
774	948	949	970	969	2
775	949	950	971	970	2
776	950	951	972	971	2
777	951	952	973	972	2
778	952	953	974	973	2
779	953	954	975	974	2
780	954	955	976	975	2
781	955	956	977	976	2
782	956	957	978	977	2
783	957	958	979	978	2
784	958	959	980	979	2
785	959	960	981	980	2
786	961	962	983	982	2
787	962	963	984	983	2
788	963	964	985	984	2
789	964	965	986	985	2
790	965	966	987	986	2
791	966	967	988	987	2
792	967	968	989	988	2
793	968	969	990	989	2
794	969	970	991	990	2
795	970	971	992	991	2
796	971	972	993	992	2
797	972	973	994	993	2
798	973	974	995	994	2
799	974	975	996	995	2
800	975	976	997	996	2
801	976	977	998	997	2
802	977	978	999	998	2
803	978	979	1000	999	2
804	979	980	1001	1000	2
805	980	981	1002	1001	2
806	982	983	1004	1003	2
807	983	984	1005	1004	2
808	984	985	1006	1005	2
809	985	986	1007	1006	2
810	986	987	1008	1007	2
811	987	988	1009	1008	2
812	988	989	1010	1009	2
813	989	990	1011	1010	2
814	990	991	1012	1011	2
815	991	992	1013	1012	2
816	992	993	1014	1013	2
817	993	994	1015	1014	2
818	994	995	1016	1015	2
819	995	996	1017	1016	2
820	996	997	1018	1017	2
821	997	998	1019	1018	2
822	998	999	1020	1019	2
823	999	1000	1021	1020	2
824	1000	1001	1022	1021	2
825	1001	1002	1023	1022	2
826	1024	1025	1034	1033	2
827	1025	1026	1035	1034	2
828	1026	1027	1036	1035	2
829	1027	1028	1037	1036	2
830	1028	1029	1038	1037	2
831	1029	1030	1039	1038	2
832	1030	1031	1040	1039	2
833	1031	1032	1041	1040	2
834	1033	1034	1043	1042	2
835	1034	1035	1044	1043	2
836	1035	1036	1045	1044	2
837	1036	1037	1046	1045	2

838	1037	1038	1047	1046	2
839	1038	1039	1048	1047	2
840	1039	1040	1049	1048	2
841	1040	1041	1050	1049	2
842	1042	1043	1052	1051	2
843	1043	1044	1053	1052	2
844	1044	1045	1054	1053	2
845	1045	1046	1055	1054	2
846	1046	1047	1056	1055	2
847	1047	1048	1057	1056	2
848	1048	1049	1058	1057	2
849	1049	1050	1059	1058	2
850	1051	1052	1061	1060	2
851	1052	1053	1062	1061	2
852	1053	1054	1063	1062	2
853	1054	1055	1064	1063	2
854	1055	1056	1065	1064	2
855	1056	1057	1066	1065	2
856	1057	1058	1067	1066	2
857	1058	1059	1068	1067	2
858	1060	1061	1070	1069	2
859	1061	1062	1071	1070	2
860	1062	1063	1072	1071	2
861	1063	1064	1073	1072	2
862	1064	1065	1074	1073	2
863	1065	1066	1075	1074	2
864	1066	1067	1076	1075	2
865	1067	1068	1077	1076	2
866	1078	1079	1083	1082	2
867	1079	1080	1084	1083	2
868	1080	1081	1085	1084	2
869	1082	1083	1087	1086	2
870	1083	1084	1088	1087	2
871	1084	1085	1089	1088	2
872	1086	1087	1091	1090	2
873	1087	1088	1092	1091	2
874	1088	1089	1093	1092	2
875	1090	1091	1095	1094	2
876	1091	1092	1096	1095	2
877	1092	1093	1097	1096	2
878	1094	1095	1099	1098	2
879	1095	1096	1100	1099	2
880	1096	1097	1101	1100	2
881	1098	1099	1103	1102	2
882	1099	1100	1104	1103	2
883	1100	1101	1105	1104	2
884	1102	1103	1107	1106	2
885	1103	1104	1108	1107	2
886	1104	1105	1109	1108	2
887	1106	1107	1111	1110	2
888	1107	1108	1112	1111	2
889	1108	1109	1113	1112	2
890	1110	1111	1115	1114	2
891	1111	1112	1116	1115	2
892	1112	1113	1117	1116	2
893	1114	1115	1119	1118	2
894	1115	1116	1120	1119	2
895	1116	1117	1121	1120	2
896	1118	1119	1123	1122	2
897	1119	1120	1124	1123	2
898	1120	1121	1125	1124	2
899	1122	1123	1127	1126	2
900	1123	1124	1128	1127	2
901	1124	1125	1129	1128	2
902	1126	1127	1131	1130	2
903	1127	1128	1132	1131	2
904	1128	1129	1133	1132	2

APPENDIX B
MESH GENERATOR FORTRAN LISTING

```
C       MESH GENERATOR -- AUTOMATIC MESH GENERATION FOR FILE
C       'MESH.IN' -- MESHES ALL AREAS IN THE FILE
C
C       IN THIS CODE 'X' COORDINATES ARE THE 'Z' AXIS
C       ............ 'Y' COORDINATES ARE THE 'R' AXIS
C
        CHARACTER*1 RIN
        DIMENSION ME(8)
C
1       WRITE (*,*)
        WRITE (*,*) 'CHOOSE:'
        WRITE (*,*) '1 - MESH NEW "MESH.IN" FILE'
        WRITE (*,*) '2 - MERGE PERIMETER NODES'
        WRITE (*,*) '3 - CONDENSE OUT UNUSED NODE NUMBERS'
        WRITE (*,*) '4 - (NOT USED)'
        WRITE (*,*) '5 - END (WRITE OUTPUT FILES)'
        WRITE (*,*) '  - INCLUDING F.E. COMPATIBLE INPUT FILE'
        WRITE (*,*)
        READ (*,*) ICHOICE
        WRITE (*,*)
        IF (ICHOICE.EQ.2) GO TO 1500
        IF (ICHOICE.EQ.3) GO TO 2500
        IF (ICHOICE.EQ.4) GO TO 3000
        IF (ICHOICE.EQ.5) GO TO 4000
        OPEN (1,FILE='MESH.IN',STATUS="OLD", FORM="FORMATTED",
     *  RECL=1)
C
C       READ ENTIRE INPUT FILE AND CREATE A FORMATTED SCRATCH FILE
C       POINTS
        OPEN (4,FILE='INPUT',ACCESS="DIRECT",FORM="FORMATTED",
     *  RECL=80,STATUS="NEW")
        OPEN (7,FILE='PLOT.IN',STATUS="NEW")
        NUMP=0
7       FORMAT (1X,I12,F15.6,F15.6)
8       FORMAT (1X,5(I12),I5)
        NUML=0
        NUMS=0
        NUMA=0
C       FIND ALL POINTS
20      FORMAT (A1,I5,F15.6,F15.6)
10      READ (1,*) RIN
        IF (RIN.EQ.'*') GO TO 30
        IF (RIN.NE.'P') GO TO 10
        WRITE (*,*) RIN
        BACKSPACE (1)
        X=0.0
        Y=0.0
        N=0
        READ (1,*) RIN,N,X,Y
        WRITE (*,*) RIN,N,X,Y
        IF (N.GT.NUMP) NUMP=N
        WRITE (4,20,REC=N) RIN,N,X,Y
        GO TO 10
C       FIND ALL LINES
30      REWIND (1)
40      FORMAT (A1,5(I5))
41      FORMAT (A1,7(I5))
45      READ (1,*) RIN
        IF (RIN.EQ.'*') GO TO 50
        IF (RIN.NE.'L') GO TO 45
        WRITE (*,*) RIN
        BACKSPACE (1)
        N=0
        IC=0
        READ (1,*) RIN,N,I1,I2,ND,IC
        WRITE (*,*) RIN,N,I1,I2,ND,IC
        IF (N.GT.NUML) NUML=N
        IREC=NUMP+N
        WRITE (4,40,REC=IREC) RIN,N,I1,I2,ND,IC
        GO TO 45
C       FIND ALL SPLINES
50      REWIND (1)
60      FORMAT (A1,6(I5))
65      READ (1,*) RIN
        IF (RIN.EQ.'*') GO TO 70
        IF (RIN.NE.'S') GO TO 65
        WRITE (*,*) RIN
        BACKSPACE (1)
        N=0
        L1=0
        L2=0
        L3=0
        L4=0
        L5=0
        READ (1,*) RIN,N,L1,L2,L3,L4,L5
        WRITE (*,*) RIN,N,L1,L2,L3,L4,L5
        IF (N.GT.NUMS) NUMS=N
        IREC=NUMP+NUML+N
        WRITE (4,60,REC=IREC) RIN,N,L1,L2,L3,L4,L5
        GO TO 65
C       READ ALL AREAS
70      REWIND (1)
75      READ (1,*) RIN
```

```
      IF (RIN.EQ.'*') GO TO 150
      IF (RIN.NE.'A') GO TO 75
      WRITE (*,*) RIN
      BACKSPACE (1)
      N=0
      J1=0
      J2=0
      J3=0
      J4=0
      READ (1,*) RIN,N,J1,J2,J3,J4,MTYPE,NMATERIAL
      WRITE (*,*) RIN,N,J1,J2,J3,J4,MTYPE,NMATERIAL
      IF (N.GT.NUMA) NUMA=N
      IREC=NUMP+NUML+NUMS+N
      WRITE (4,41,REC=IREC) RIN,N,J1,J2,J3,J4,MTYPE,NMATERIAL
      GO TO 75
C
C     WRITE A POINT AND LINE PLOT INPUT FILE FOR MESHPLOT
151   FORMAT (A1,I12,F15.6,F15.6,I9,I9)
150   WRITE (7,*) 'N',NUMP,NUML,0,0,0
      DO 155 I=1,NUMP
      IREC=I
      READ (4,20,REC=IREC) RIN,N,X,Y
      WRITE (7,151) RIN,N,X,Y,0,0
155   CONTINUE
      DO 160 I=1,NUML
      IREC=NUMP+I
      READ (4,40,REC=IREC) RIN,N,I1,I2,I3,I4
      WRITE (7,*) RIN,N,I1,I2,I3,I4
160   CONTINUE
C     DO 165 I=1,NUMS
C     IREC=NUMP+NUML+I
C     READ (4,60,REC=IREC) RIN,N,L1,L2,L3,L4,L5
C     WRITE (7,*) RIN,N,L1,L2,L3,L4,L5
C165    CONTINUE
C     DO 170 I=1,NUMA
C     IREC=NUMP+NUML+NUMS+I
C     READ (4,41,REC=IREC) RIN,N,J1,J2,J3,J4,IM,NM
C     WRITE (7,*) RIN,N,J1,J2,J3,J4
C170    CONTINUE
      CLOSE (7)
C     STARTING NODE NUMBER
      IN=0
      CLOSE (1)
      OPEN (1,FILE='MERGE',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=5,STATUS="NEW")
      OPEN (2,FILE='NODESL',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=43,STATUS="NEW")
C     STARTING ELEMENT NUMBER
      IE=0
      OPEN (3,FILE='ELEML',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=66,STATUS="NEW")
C
      DO 1000 IA=1,NUMA
      IREC=NUMP+NUML+NUMS+IA
      N=0
      JP1=0
      JP2=0
      JP3=0
      JP4=0
      MTYPE=0
      READ (4,41,REC=IREC) RIN,N,JP1,JP2,JP3,JP4,MTYPE,NMAT
      IF (N.EQ.0) GO TO 1000
      WRITE (*,*) RIN,N,JP1,JP2,JP3,JP4,MTYPE,NMAT
      ND1=0
      ND2=0
C
      WRITE (*,*) 'CALL EDGEMESH K=1'
C     FIND AND MESH LINE, RADIUS, OR SPLINE CONNECTING JP1 TO JP2
      CALL EDGEMESH (1,JP1,JP2,ND1,ND2,IN,NUMP,NUML,NUMS)
      WRITE (*,*) 'CALL EDGEMESH K=2'
C     FIND AND MESH LINE, RADIUS, OR SPLINE CONNECTING JP2 TO JP3
      CALL EDGEMESH (2,JP2,JP3,ND1,ND2,IN,NUMP,NUML,NUMS)
C     FIND AND MESH LINE, RADIUS, OR SPLINE CONNECTING JP3 TO JP4
      CALL EDGEMESH (3,JP4,JP3,ND1,ND2,IN,NUMP,NUML,NUMS)
C     FIND AND MESH LINE, RADIUS, OR SPLINE CONNECTING JP4 TO JP1
      CALL EDGEMESH (4,JP1,JP4,ND1,ND2,IN,NUMP,NUML,NUMS)
C
C     ASSIGN NODES TO ELEMENTS FOR THIS AREA
C
      DO 250 KE=1,ND2
      DO 200 JE=1,ND1
      IE=IE+1
      N1=IN+JE+((ND1+1)*(KE-1))
      N2=N1+1
      N3=N2+ND1+1
      N4=N3-1
      WRITE (3,8,REC=IE) IE,N1,N2,N3,N4,NMAT
200   CONTINUE
250   CONTINUE
C
C     FIND X,Y COORDINATES FOR INTERIOR NODES FOR THIS AREA
C
      NN=IN+1+ND1+1
```

```
      DO 400 I=1,(ND2-1)
      NE1=IN+1+ND1+(1+ND1)*I
      NS1=NE1-ND1
      READ (2,7,REC=NE1) NX,XE1,YE1
      WRITE (*,*) NX,XE1,YE1
      READ (2,7,REC=NS1) NX,XS1,YS1
      DX=(XE1-XS1)/ND1
      DO 300 J=1,(ND1-1)
      IF (MTYPE.EQ.2) THEN
      NE2=IN+(ND1+1)*(ND2+1)-ND1+J
      NS2=IN+1+J
      READ (2,7,REC=NE2) NY,XE2,YE2
      READ (2,7,REC=NS2) NY,XS2,YS2
      DY=(YE2-YS2)/ND2
      Y=YS2+DY*I
      ENDIF
      IF (MTYPE.EQ.1) THEN
      DY=(YE1-YS1)/ND1
      Y=YS1+DY*J
      ENDIF
      NN=NN+1
      X=XS1+DX*J
      WRITE (2,7,REC=NN) NN,X,Y
300   CONTINUE
      NN=NN+2
400   CONTINUE
      IN=IN+(ND1+1)*(ND2+1)
1000  CONTINUE
1001  FORMAT (I5)
      WRITE (1,1001) 0
      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)
      CLOSE (5)
      CLOSE (6)
      OPEN (1,FILE='TOTALS',STATUS="NEW")
      WRITE (1,*) IN,IE
      CLOSE (1)
      GO TO 1
C
C     MERGE NODES ROUTINE
1500  OPEN (1,FILE='TOTALS')
      READ (1,*) IN,IE
      CLOSE (1)
      OPEN (2,FILE='NODESL',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=43)
      OPEN (3,FILE='ELEML',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=66)
      OPEN (4,FILE='MERGE',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=5)
C     CREATE A NODE-ELEMENT FILE
      OPEN (5,FILE='NOEL',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=45)
1510  FORMAT (9(I5))
      CALL NOEL (IN,IE)
C
C     MERGE OUT NODES ON COMMON PERIMETER LINES
C
1610  FORMAT (A2)
1611  FORMAT (I5)
      JREC=1
1615  KREC=JREC
      READ (4,1611,REC=KREC) ISTAR
      WRITE (*,*) 'ISTAR',ISTAR
      IF (ISTAR.EQ.0) GO TO 2400
      READ (4,1611,REC=KREC) L1
      WRITE (*,*) 'L1=',L1
      KREC=KREC+1
      READ (4,1611,REC=KREC) NNL1
      KREC=KREC+NNL1+1
      JREC=KREC
      IF (L1.EQ.-1) GO TO 1615
C     FIND IF LINE L1 REPEATS IN THIS FILE AND MERGE OUT NODES
1620  READ (4,1611,REC=KREC) ISTAR
      WRITE (*,*) 'ISTAR',ISTAR
      IF (ISTAR.EQ.0) GO TO 1615
      READ (4,1611,REC=KREC) L2
      WRITE (*,*) 'L2=',L2
      MREC=KREC
      KREC=KREC+1
      READ (4,1611,REC=KREC) NNL2
      IF (L1.NE.L2) GO TO 1650
C     WE HAVE A MATCH
      KREC=KREC+1
      WRITE (4,1611,REC=MREC) -1
C     REPLACE L2 NODES WITH L1 NODES IN ELEMENT-NODE FILE 'ELEML'
      LREC1=JREC-NNL1
      DO 1625 I=1,NNL1
      READ (4,1611,REC=LREC1) N1
      READ (4,1611,REC=KREC) N2
      LREC1=LREC1+1
      KREC=KREC+1
```

```
      WRITE (*,*) N1,N2
      READ (5,1510,REC=N2) JN2,ME(1),ME(2),ME(3),ME(4),ME(5),ME(6),
     * ME(7),ME(8)
      DO 1624 J=1,8
      IF (ME(J).EQ.0) GO TO 1624
      NREC=ME(J)
      READ (3,8,REC=NREC) N,J1,J2,J3,J4,NMAT
      IF (N2.EQ.J1) WRITE (3,8,REC=NREC) N,N1,J2,J3,J4,NMAT
      IF (N2.EQ.J2) WRITE (3,8,REC=NREC) N,J1,N1,J3,J4,NMAT
      IF (N2.EQ.J3) WRITE (3,8,REC=NREC) N,J1,J2,N1,J4,NMAT
      IF (N2.EQ.J4) WRITE (3,8,REC=NREC) N,J1,J2,J3,N1,NMAT
C     REPLACE N2 WITH 0 IN 'NODESL' FILE
      WRITE (2,7,REC=N2) 0,0,0
1624  CONTINUE
1625  CONTINUE
      GO TO 1620
1650  KREC=KREC+NNL2+1
1660  GO TO 1620
2400  CONTINUE
C     FINISHED
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)
      CLOSE (5)
      GO TO 1
C
C     CONDENSE NODE NUMBERS ROUTINE
2500  OPEN (1,FILE='TOTALS')
      READ (1,*) IN,IE
      OPEN (2,FILE='NODESL',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=43)
      OPEN (3,FILE='ELEML',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=66)
      OPEN (5,FILE='NOEL',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=45)
      CALL NOEL (IN,IE)
      ID=0
      DO 2600 IR=1,IN
      READ (2,7,REC=IR) N,X,Y
      WRITE (*,*) N,X,Y
      IF (N.EQ.0) ID=ID+1
      IF (N.EQ.0) GO TO 2600
      IF (ID.EQ.0) GO TO 2600
      IW=IR-ID
      READ (5,1510,REC=IR) JN2,ME(1),ME(2),ME(3),ME(4),ME(5),ME(6),
     * ME(7),ME(8)
      N=IR-ID
      WRITE (2,7,REC=IW) N,X,Y
      WRITE (2,7,REC=IR) 0,0,0
      DO 2550 J=1,8
      IF (ME(J).EQ.0) GO TO 2550
      NREC=ME(J)
      READ (3,8,REC=NREC) JE,J1,J2,J3,J4,JMAT
      IF (IR.EQ.J1) WRITE (3,8,REC=NREC) JE,N,J2,J3,J4,JMAT
      IF (IR.EQ.J2) WRITE (3,8,REC=NREC) JE,J1,N,J3,J4,JMAT
      IF (IR.EQ.J3) WRITE (3,8,REC=NREC) JE,J1,J2,N,J4,JMAT
      IF (IR.EQ.J4) WRITE (3,8,REC=NREC) JE,J1,J2,J3,N,JMAT
2550  CONTINUE
2600  CONTINUE
      IN=IN-ID
      REWIND (1)
      WRITE (1,*) IN,IE
      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (5)
      GO TO 1
C
C     RENUMBER NODES, AND ELEMENTS (BANDWIDTH) ROUTINE
3000  CONTINUE
      GO TO 1
C
C     WRITE OUTPUTFILES COMPATIBLE WITH MESHPLOT
4000  OPEN (5,FILE='NODES',STATUS="NEW")
      OPEN (6,FILE='ELEMENTS',STATUS="NEW")
      OPEN (2,FILE='NODESL',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=43)
      OPEN (3,FILE='ELEML',ACCESS="DIRECT",FORM="FORMATTED",
     * RECL=66)
      OPEN (4,FILE='FE.IN')
4001  FORMAT (' ET,1,42,0,0,1')
4002  FORMAT (' EX,',I3,',')
4003  FORMAT (' NUXY,',I3,',')
4004  FORMAT (' DENS,',I3,',')
4005  FORMAT (' MAT,',I3)
      OPEN (1,FILE='TOTALS')
      READ (1,*) IN,IE
      CLOSE (1)
      DO 4010 I=1,IN
      READ (2,7,REC=I) N,X,Y
      WRITE (5,*) ' ',N,X,Y
C     F.E. USES Y=Z AND X=R
      WRITE (4,*) ' N,',N,',',Y,',',X
```

```
4010    CONTINUE
        WRITE (4,4001)
        NNMAT=0
        DO 4020 I=1,IE
        READ (3,8,REC=I) N,N1,N2,N3,N4,NMAT
        WRITE (6,*) ' ',N,N1,N2,N3,N4,NMAT
        IF (NMAT.EQ.NNMAT) GO TO 4015
        NNMAT=NMAT
        WRITE (4,4002) NNMAT
        WRITE (4,4003) NNMAT
        WRITE (4,4004) NNMAT
        WRITE (4,4005) NNMAT
4015    WRITE (4,*) ' E,',N1,',',N2,',',N3,',',N4
4020    CONTINUE

5000    END

        SUBROUTINE NOEL (IN,IE)
1       FORMAT (9(I5))
2       FORMAT (1X,5(I12),I5)
        DO 10 I=1,IN
10      WRITE (5,1,REC=I) I,0,0,0,0,0,0,0,0
        DO 20 I=1,IE
        READ (3,2,REC=I) N,N1,N2,N3,N4,NMAT
        CALL ENTER (N1,I)
        CALL ENTER (N2,I)
        CALL ENTER (N3,I)
        CALL ENTER (N4,I)
20      CONTINUE
        RETURN
        END

        SUBROUTINE ENTER (K,J)
1       FORMAT (9(I5))
        READ (5,1,REC=K) I,M1,M2,M3,M4,M5,M6,M7,M8
        IF (M1.EQ.0) WRITE (5,1,REC=K) I,J,M2,M3,M4,M5,M6,M7,M8
        IF (M1.EQ.0) GO TO 10
        IF (M2.EQ.0) WRITE (5,1,REC=K) I,M1,J,M3,M4,M5,M6,M7,M8
        IF (M2.EQ.0) GO TO 10
        IF (M3.EQ.0) WRITE (5,1,REC=K) I,M1,M2,J,M4,M5,M6,M7,M8
        IF (M3.EQ.0) GO TO 10
        IF (M4.EQ.0) WRITE (5,1,REC=K) I,M1,M2,M3,J,M5,M6,M7,M8
        IF (M4.EQ.0) GO TO 10
        IF (M5.EQ.0) WRITE (5,1,REC=K) I,M1,M2,M3,M4,J,M6,M7,M8
        IF (M5.EQ.0) GO TO 10
        IF (M6.EQ.0) WRITE (5,1,REC=K) I,M1,M2,M3,M4,M5,J,M7,M8
        IF (M6.EQ.0) GO TO 10
        IF (M7.EQ.0) WRITE (5,1,REC=K) I,M1,M2,M3,M4,M5,M6,J,M8
        IF (M7.EQ.0) GO TO 10
        IF (M8.EQ.0) WRITE (5,1,REC=K) I,M1,M2,M3,M4,M5,M6,M7,J
10      RETURN
        END

        SUBROUTINE EDGEMESH (K,JJ1,JJ2,ND1,ND2,IIN,NUMP,NUML,NUMS)
C       'K' = EDGE NUMBER; 'IIN' = ENDING NODE NUMBER OF LAST AREA
        CHARACTER*1 RIN
        DIMENSION L(5),IP1(5),IP2(5),IND(5),IC(5)

1       FORMAT (I5)
2       FORMAT (A1,I5,F15.6,F15.6)
3       FORMAT (A1)
4       FORMAT (A1,5(I5))
6       FORMAT (A1,6(I5))
        GO TO 20
C
C       LOOK FOR A SPLINE CONNECTING JJ1 AND JJ2
C
7       IF (NUMS.EQ.0) GO TO 100
        IREC=NUMP+NUML
10      IREC=IREC+1
        IF (IREC.GT.(NUMP+NUML+NUMS)) GO TO 100
        ICHECK=0
        READ (4,6,REC=IREC) RIN,N,L(1),L(2),L(3),L(4),L(5)
        WRITE (*,*) RIN,N,L(1),L(2),L(3),L(4),L(5)
        JF=5
        IF (L(2).EQ.0) GO TO 10
        IF (L(5).EQ.0) JF=4
        IF (L(4).EQ.0) JF=3
        IF (L(3).EQ.0) JF=2
16      JREC=NUMP+L(1)
        READ (4,4,REC=JREC) RIN,L(1),IP1(1),IP2(1),IND(1),IC(1)
        WRITE (*,*) RIN,L(1),IP1(1),IP2(1),IND(1),IC(1)
        IF (IP2(1).EQ.JJ1) THEN
        IRP=IP1(1)
        IP1(1)=IP2(1)
        IP2(1)=IRP
        ENDIF
        IF (IP1(1).EQ.JJ1) GO TO 17
        IF (ICHECK.EQ.1) GO TO 10
C       REVERSE SPLINE ORDER AND THEN CHECK
        L1=L(5)
        L2=L(4)
        L3=L(3)
```

```
      L4=L(2)
      L5=L(1)
      IF (JF.EQ.5) THEN
      L(1)=L1
      L(2)=L2
      L(3)=L3
      L(4)=L4
      L(5)=L5
      ENDIF
      IF (JF.EQ.4) THEN
      L(1)=L2
      L(2)=L3
      L(3)=L4
      L(4)=L5
      L(5)=L1
      ENDIF
      IF (JF.EQ.3) THEN
      L(1)=L3
      L(2)=L4
      L(3)=L5
      L(4)=L1
      L(5)=L2
      ENDIF
      IF (JF.EQ.2) THEN
      L(1)=L4
      L(2)=L5
      L(3)=L1
      L(4)=L2
      L(5)=L3
      ENDIF
      ICHECK=1
      GO TO 16
17    JREC=NUMP+L(JF)
      READ (4,4,REC=JREC) RIN,L(JF),IP1(JF),IP2(JF),IND(JF),IC(JF)
      WRITE (*,*) RIN,L(JF),IP1(JF),IP2(JF),IND(JF),IC(JF)
      IF (IP1(JF).EQ.JJ2) THEN
      IRP=IP2(JF)
      IP2(JF)=IP1(JF)
      IP1(JF)=IRP
      ENDIF
      IF (IP2(JF).NE.JJ2) GO TO 10
      WRITE (*,*) 'ITS A SPLINE'
C     **** IT'S A SPLINE ****
C     READ THE INTERIOR LINES
      IF (JF.EQ.2) GO TO 13
      DO 12 I=1,JF
      IF (I.EQ.1) GO TO 11
      IF (I.EQ.JF) GO TO 11
      JREC=NUMP+L(I)
      READ (4,4,REC=JREC) RIN,L(I),IP1(I),IP2(I),IND(I),IC(I)
      WRITE (*,*) RIN,L(I),IP1(I),IP2(I),IND(I),IC(I)
      IF (IP1(I).NE.IP2(I-1)) THEN
      IRP=IP1(I)
      IP1(I)=IP2(I)
      IP2(I)=IRP
      ENDIF
11    CONTINUE
12    CONTINUE
13    CONTINUE
      ND=0
      DO 14 I=1,JF
14    ND=ND+IND(I)
      IF (K.EQ.1) THEN
      ND1=ND
      NS=IIN+1
      NIN=1
      ENDIF
      IF (K.EQ.2) THEN
      ND2=ND
      NS=IIN+(1+ND1)
      NIN=1+ND1
      ENDIF
      IF (K.EQ.3) THEN
      NS=IIN+(1+ND1)*ND2+1
      NIN=1
      ENDIF
      IF (K.EQ.4) THEN
      NS=IIN+1
      NIN=1+ND1
      ENDIF
      DO 15 I=1,JF
      WRITE (*,*) 'CALL MESH (SPLINE LINE)'
      WRITE (1,1) L(I)
      WRITE (1,1) IND(I)+1
      CALL MESH (NS,NIN,IP1(I),IP2(I),IND(I),IC(I))
      NS=NS+NIN*IND(I)
15    CONTINUE
      GO TO 100
C
C     LOOK FOR A LINE OR RADIUS
C
20    IREC=NUMP
25    IREC=IREC+1
```

```
      IF (IREC.GT.(NUMP+NUML)) GO TO 7
      READ (4,4,REC=IREC) RIN,N,IP1(1),IP2(1),IND(1),IC(1)
      WRITE (*,*) RIN,N,IP1(1),IP2(1),IND(1),IC(1)
      IF (IP1(1).EQ.JJ2) THEN
      IRP=IP1(1)
      IP1(1)=IP2(1)
      IP2(1)=IRP
      ENDIF
      IF (IP1(1).NE.JJ1) GO TO 25
      IF (IP2(1).NE.JJ2) GO TO 25
      IF (K.EQ.1) THEN
      ND1=IND(1)
      NS=IIN+1
      NIN=1
      ENDIF
      IF (K.EQ.2) THEN
      ND2=IND(1)
      NS=IIN+1+ND1
      NIN=1+ND1
      ENDIF
      IF (K.EQ.3) THEN
      NS=IIN+(1+ND1)*ND2+1
      NIN=1
      ENDIF
      IF (K.EQ.4) THEN
      NS=IIN+1
      NIN=1+ND1
      ENDIF
      WRITE (1,1) N
      WRITE (1,1) IND(1)+1
      WRITE (*,*) 'CALL MESH LINE OR RADIUS'
      CALL MESH (NS,NIN,IP1(1),IP2(1),IND(1),IC(1))
100   RETURN
      END

      SUBROUTINE MESH (NS,NIN,I1,I2,ND,JC)
      CHARACTER*1 RIN
1     FORMAT (A1,I5,F15.6,F15.6)
2     FORMAT (1X,I12,F15.6,F15.6)
3     FORMAT (I5)
      IREC=I1
      READ (4,1,REC=IREC) RIN,N,X1,Y1
      WRITE (*,*) RIN,N,X1,Y1
      IREC=I2
      READ (4,1,REC=IREC) RIN,N,X2,Y2
      IF (JC.NE.0) THEN
      IREC=JC
      READ (4,1,REC=IREC) RIN,N,XC,YC
      ENDIF
      I=NS
      IF (JC.NE.0) GO TO 20
      DX=(X2-X1)/ND
      XS=X1
      DY=(Y2-Y1)/ND
      YS=Y1
      DO 10 J=0,ND
      XI=XS+DX*J
      YI=YS+DY*J
      WRITE (2,2,REC=I) I,XI,YI
      WRITE (1,3) I
      I=I+NIN
10    CONTINUE
      GO TO 100
20    R=((X1-XC)**2+(Y1-YC)**2)**.5
      DX1=ABS(X1-XC)
      DX2=ABS(X2-XC)
      DY1=ABS(Y1-YC)
      DY2=ABS(Y2-YC)
      PI=3.14159
      IF (X1.GT.XC.AND.Y1.GE.YC) T1=ATAN(DY1/DX1)
      IF (X2.GT.XC.AND.Y2.GE.YC) T2=ATAN(DY2/DX2)
      IF (X1.GT.XC.AND.Y1.LT.YC) T1=2.0*PI-ATAN(DY1/DX1)
      IF (X2.GT.XC.AND.Y2.LT.YC) T2=2.0*PI-ATAN(DY2/DX2)
      IF (X1.LT.XC.AND.Y1.GE.YC) T1=PI-ATAN(DY1/DX1)
      IF (X2.LT.XC.AND.Y2.GE.YC) T2=PI-ATAN(DY2/DX2)
      IF (X1.LT.XC.AND.Y1.LT.YC) T1=PI+ATAN(DY1/DX1)
      IF (X2.LT.XC.AND.Y2.LT.YC) T2=PI+ATAN(DY2/DX2)
      IF (X1.EQ.XC.AND.Y1.GT.YC) T1=PI/2.0
      IF (X2.EQ.XC.AND.Y2.GT.YC) T2=PI/2.0
      IF (X1.EQ.XC.AND.Y1.LT.YC) T1=PI+PI/2.0
      IF (X2.EQ.XC.AND.Y2.LT.YC) T2=PI+PI/2.0
      IF (T1.GE.0.0.AND.T1.LE.(PI/2.0).AND.T2.GE.(PI+PI/2.0))
     * T2=2.0*PI-T2
      IF (T2.GE.0.0.AND.T2.LE.(PI/2.0).AND.T1.GE.(PI+PI/2.0))
     * T2=T2+2.0*PI
      DT=(T2-T1)/ND
      DO 30 J=0,ND
      XI=XC+R*COS(T1+J*DT)
      YI=YC+R*SIN(T1+J*DT)
      WRITE (2,2,REC=I) I,XI,YI
      WRITE (1,3) I
      I=I+NIN
30    CONTINUE
100   RETURN
      END
```